Hagen Hirschmann

Korpuslinguistik

Eine Einführung

Lösungen der Arbeitsaufgaben zum Buch

J. B. Metzler Verlag

Arbeitsaufgabe 2.2.2

- Lesen Sie anhand der vorangegangenen Informationen die nach Teilsätzen tokenisierte Datei (Aufgabe 2. aus Kap. 2.2.25; https://bit.ly/2Om4RdS) in EXMARaLDA ein, so dass die Tokensegmentierung korrekt wiedergegeben wird und die Separatoren in der Ergebnisdatei gelöscht sind.
- Speichern Sie das Ergebnis als EXMARaLDA-Datei namens »EXMARaLDA_Import_tokenisiert_3.exb« ab.

Lösung

Lösungsdatei »EXMARaLDA_Import_tokenisiert_3.exb«: https://bit.ly/2I8wplO

Arbeitsaufgabe 2.2.3

- Bilden Sie die in Abb. 2.4 gezeigten Annotationen für die Schrift- und Textmerkmale des NABU-Artikels auf der Webseite https://bit.ly/2OmG4q0 nach (siehe Abb. 2.3), damit Sie sich mit der manuellen Annotation im EXMARaLDA-Partitureditor vertraut machen.

Lösung

Lösungsdatei »EXMARaLDA_Sumpfschildkröten_Text_und_Schriftmerkmale.exb«: https://bit.ly/2TTutzG

Arbeitsaufgabe 2.2.4

- Lesen Sie die Datei, die unter der Webadresse https://bit.ly/2FrgXPY verfügbar ist, in EXMARaLDA ein.
  - Nutzen Sie dazu die »File«-»Import...«-Funktion und wählen Sie das Datenformat »Plain text file (*.txt)«.
  - Geben Sie anschließend unter der Option »Split at regular expression:« ein Leerzeichen ein.
    (Die importierten Daten entsprechen der unter der Webadresse https://bit.ly/2CuMqik verfügbaren Datei. Diese kann alternativ über die Funktion »File« > »Open« im Partitureditor oder über das Ausführen der Datei mit dem Programm EXMARaLDA bzw. »PartiturEditor*.exe« bearbeitet werden.)
- Hierbei handelt es sich um einen Text mit gewissen orthographischen Fehlern wie kleingeschriebenen Nomina. Werden diese nicht normgerechten Formen im Korpus ohne Normalisierung weiterverarbeitet. können sie später schwer aufgefunden werden und bergen ein hohes Risiko, bei automatischen Verarbeitungsverfahren falsch kategorisiert zu werden. Dies heißt ausdrücklich nicht, dass solche Normverstöße nicht selber von linguistischem Interesse sein können und gänzlich aus den Korpusdaten eliminiert werden sollen. Deshalb ist es wichtig, im Korpus viele unterschiedliche Beschreibungsebenen zu haben, die in ihrer Summe möglichst viele Eigenschaften der im Korpus verarbeiteten Primärdaten abbilden.

  - Fügen Sie in diesem Sinne eine Annotationsebene »Norm« hinzu (»Insert tier«), übernehmen Sie bei der Erzeugung dieser Ebene die Elemente der Ebene »TXT« (»Copy events …«) und bearbeiten Sie die Annotationsebene »Norm«, ohne die Text-Ebene (»TXT«) zu verändern, so dass auf der Beschreibungsebene »Norm« ausschließlich Formen nach der Standardschreibung (so, wie man die Formen im Lexikon suchen würde) stehen. Es kann nicht nur sein, dass Sie hierfür den Inhalt in den Zellen verändern müssen, sondern auch, dass Sie ggf. Zellen auftrennen bzw. Zellen hinzufügen (Funktion: »Split« oder Zellen zusammenführen müssen (Funktion: »Merge«).
    Achten Sie jedoch darauf, dass am rechten Rand einer jeden Zelle das Leerzeichen erhalten bleibt. Dann können Sie anschließend die gesamte Spur »Norm« markieren, kopieren, und den Inhalt in eine Textdatei hineinkopieren (ohne Leerzeichen erhalten Sie hierbei eine fortlaufende Zeichenkette).
- Speichern Sie das Ergebnis unter dem Namen »Normalisieren_normalisiert.exb«. Speichern Sie auch eine Textdatei »Normalisieren_normalisiert.txt«, indem Sie den Inhalt der »Norm«-Spur in eine Textdatei kopieren und diese speichern.

Lösung

Lösungsdatei »Normalisieren_normalisiert.exb«: https://bit.ly/2OGnyJ8
Lösungsdatei »Normalisieren_normalisiert.txt«: https://bit.ly/2IaFR8j

Arbeitsaufgabe 2.2.5

- Beziehen Sie von der Internetadresse https://bit.ly/2CxEr3R einen untokenisierten Text. Hierbei handelt es sich um den bereits normalisierten Text aus der Normalisierungsaufgabe in Kap. 2.2.4.
- Tokenisieren Sie den Text innerhalb der Datei nach den folgenden Maßgaben:

Nr.	Token-Definition	Tokenisierungsseparator
1.	Ein Token ist genau ein Wort oder ein Satzzeichen.	Die Tokengrenze wird durch Absätze angezeigt. Leerzeichen sind nur bei wortinternem Gebrauch legitim.
2.	Ein Token ist genau ein Teilsatz.	Die Tokengrenze wird mit Rautenzeichen (»#«) angezeigt. Leerzeichen markieren Wortgrenzen und gehören somit zu den einzelnen Token (sie sind Subtoken).
3.	Ein Token ist genau ein Morphem oder ein Satzzeichen.	Die Tokengrenze wird durch @-Zeichen markiert. Leerzeichen sind kein legitimes Zeichen im Korpus.

- Speichern Sie die jeweils bearbeitete Datei gesondert mit dem Zusatz »_tokenisiert_1« bzw. »_tokenisiert_2« und »_tokenisiert_3«.

Lösung

Lösungsdatei »Tokenisieren_ tokenisiert_1.txt«: https://bit.ly/2UgcqZy
Lösungsdatei »Tokenisieren_ tokenisiert_2.txt«: https://bit.ly/2YO0U6c
Lösungsdatei »Tokenisieren_ tokenisiert_3.txt«: https://bit.ly/2WGmY0N

Arbeitsaufgabe 2.2.6.2

Annotieren Sie die nach Wörtern und Satzzeichen mittels Absätzen tokenisierte Datei (Aufgabe 1. aus Kap. 2.2.5; https://bit.ly/2Oge2MR).

- Verwenden Sie dabei das unter https://bit.ly/1KWyVVM verfügbare Tagset. Eine offline verwendbare Version finden Sie hier: https://bit.ly/2TjSWhq.
- Die Annotationsrichtlinien zum STTS finden Sie hier: https://bit.ly/2TVWURK.
- Vergeben Sie die Annotationskürzel (Tags) des STTS entweder, indem Sie den Token in der Textdatei einen Tabulatorabstand anfügen und dann manuell die STTS-Werte ausschließlich in Großbuchstaben hinzufügen, oder kopieren Sie den Dateiinhalt in eine Arbeitsmappe des Programms LibreOffice Calc oder Microsoft Excel und fügen Sie den Token in der jeweils rechts danebenliegenden Tabellenspalte die passenden STTS-Werte hinzu.
- Speichern Sie die annotierte Datei als Textdatei unter dem Namen »Tokenisiert_1_getaggt.txt«. Wenn Sie im Tabellenprogramm gearbeitet haben, kopieren Sie die zwei Spalten in eine Textdatei zurück oder speichern Sie das Ergebnis als .csv-Datei mit tabulaturgetrennten Spalten. Sie können anschließend die Dateiendung in .txt umbenennen und die Datei in einem einfachen Texteditor öffnen.

Lösung

Lösungsdatei »Tokenisiert_1_getaggt.txt«: https://bit.ly/2FLoS9M

Arbeitsaufgabe 2.2.6.3

Lemmatisieren Sie die nach Wörtern und Satzzeichen mittels Absätzen tokenisierte Datei (Aufgabe 1. aus Kap. 2.2.5; https://bit.ly/2Oge2MR).

- Vergeben Sie dabei jedem Token genau eine Grundform. Entspricht die Form des Tokens bereits der Grundform, schreiben Sie diese Form noch einmal hin. Achten Sie auch auf die Groß- und Kleinschreibung, gerade an Satzanfängen.
- Vergeben Sie die Annotationskürzel (Tags) des STTS entweder, indem Sie den Token in der Textdatei einen Tabulatorabstand anfügen und dann manuell die STTS-Werte ausschließlich in Großbuchstaben hinzufügen, oder kopieren Sie den Dateiinhalt in eine Arbeitsmappe des Programms LibreOffice Calc oder Microsoft Excel und fügen Sie den Token in der jeweils rechts danebenliegenden Tabellenspalte die passenden STTS-Werte hinzu.
- Speichern Sie die annotierte Datei als Textdatei unter dem Namen »Tokenisiert_1_lemmatisiert.txt«. Wenn Sie im Tabellenprogramm gearbeitet haben, kopieren Sie die zwei Spalten in eine Textdatei zurück oder spei-

chern Sie das Ergebnis als .csv-Datei mit tabulaturgetrennten (Trennzeichen-getrennten) Spalten. Sie können anschließend die Dateiendung in .txt umbenennen und die Datei in einem einfachen Texteditor öffnen.

Lösung

Lösungsdatei »Tokenisiert_1_lemmatisiert.txt«: https://bit.ly/2FVxITw

Arbeitsaufgabe 2.2.6.4

- Taggen Sie die Datei, die zuvor manuell nach Wortarten und Lemmata annotiert werden sollte, mit AntConc. Die Datei können Sie unter https://bit.ly/2CxEr3R beziehen. Verwenden Sie die oben stehenden Handlungsanweisungen für die Verwendung des Taggers.
- Überprüfen Sie das Taggingergebnis, indem Sie die von Tagger ausgegebene Datei Tokenisieren_tagged.txt mit den Ergebnissen des manuellen Wortartentaggings (Tokenisiert_1_getaggt.txt) und des manuellen Lemmatisierens (Tokenisiert_1_lemmatisiert.txt) vergleichen.
  Um dies computerunterstützt zu tun, können Sie aus den manuell erstellten Dateien eine Datei mit einer dreispaltigen Tabelle nach dem Vorbild der von TagAnt ausgegebenen Datei erzeugen und anschließend die manuelle und die automatisch erzeugte Analyse mit einem Programm vergleichen: eine passende Funktion besitzt Microsoft Word; ein frei erhältliches Vergleichsprogramm ist KDiff3 (https://kdiff3.sourceforge.net/ bzw. https://sourceforge.net/projects/kdiff3/files/kdiff3/).

Lösung

- Lösungsdatei »Tagging_Wortarten_Vergleich.txt«: https://bit.ly/2FUXvv2
  Die beiden im Forum verwendeten Namen werden vom Tagger nicht als Eigennamen erkannt. Das Kürzel *MfG* hingegen wird als Eigenname interpretiert. Das satzinitiale imperativische *Schau* wird fälschlicherweise als Nomen erkannt. Der Modalpartikel (STTS: ADV) *doch* wird der Wert für Konjunktionen (STTS: KON) zugewiesen. Insgesamt ist die Analyse des Taggers ziemlich akkurat (siehe Kap. 2.5.3 für detaillierte Evaluationen).
- Lösungsdatei »Tagging_Lemmatisierung_Vergleich.txt«: https://bit.ly/2uHVo7k
  Die Lemmatisierung des Taggers unterscheidet sich vor allem in der Behandlung von Artikelwörtern: In der verwendeten Taggerversion werden der bestimmte Artikel, der unbestimmte Artikel sowie der Indefinitartikel *jeder* mit der weiblichen Nominativform lemmatisiert. Der Indefinitartikel *alle* wird pluralisch lemmatisiert. Die Kardinalzahl *1.000* wird (wie andere Kardinalzahlen) als »@card@« lemmatisiert. Ein ›echter‹ Lemmatisierungsfehler ist das satzinitiale imperativische *Schau*, welches als das Nomen *Schau* lemmatisiert wird (in Übereinstimmung mit der Wortartenanalyse).

Arbeitsaufgabe 2.2.6.5

- Überführen Sie die annotierte Datei, die Sie unter der Webadresse https://bit.ly/2Zo9IDQ herunterladen können, in das dreispaltige TreeTagger-Format:
  - Ausgangsformat:
    Lieber@ADJA@lieb Nutzer@NN@Nutzer ...
  - Zielformat:
    Lieber ADJA lieb
    Nutzer NN Nutzer
    ...
- Sie erreichen dies, indem Sie mit einem funktionsreicheren Texteditor (wie Notepad++, https://notepad-plus-plus.org/) folgende Ersetzungen durchführen:
  - Ersetze @-Zeichen (geben Sie dieses in das »Suchen«-Feld ein) mit Tabulaturabstand (Tabstopp, Trennzeichen; kopieren Sie einen solchen in das »Ersetzen«-Feld).
  - Ersetze Leerzeichen (geben Sie ein Leerzeichen in das »Suchen«-Feld ein) mit Absatz (geben Sie den regulären Ausdruck »\n« für »Absatz« in das »Ersetzen«-Feld und geben Sie an, dass Sie erweiterte Ausdrücke bzw. reguläre Ausdrücke verwenden).
- Das Ergebnis muss aussehen wie die Daten in der getaggten Datei, die Sie mit TagAnt erzeugt haben (siehe die Taggingaufgabe in Kap. 2.2.6.4) bzw. die Lösungsdatei, die unter https://bit.ly/2TnaDMY verfügbar ist).

Lösung

Lösungsdatei »Taggen_tagged_TagAnt.txt«: https://bit.ly/2TnaDMY

Arbeitsaufgaben 2.2.6.6

1. Reichern Sie die Datei, die Sie unter der Webadresse https://bit.ly/2HK6PDj beziehen können, mit flexionsmorphologischen Werten an (die ersten zwei Sätze sind bereits beispielhaft annotiert).
   - Sie können die Datei in die Tabelleneditoren von LibreOffice (oder OpenOffice) Calc oder Microsoft Office Excel einlesen.
   - Verwenden Sie hierbei die Richtlinien von Crysmann et al. 2005 (https://bit.ly/2Hwj9rL).
   - Speichern Sie das Ergebnis mit dem Zusatz »_getaggt_auto« im .xlsx-Format.
2. Taggen Sie die Datei, die Sie unter der Webadresse https://bit.ly/2CxEr3R herunterladen können, mit flexionsmorphologischen Kategorien, indem Sie in WebLicht den RFTagger darauf anwenden.
   - Lesen Sie hierfür die Datei »Tokenisieren.txt« ein (Funktion »Upload a file: Browse«).
   - Spezifizieren Sie die Sprache (wählen Sie »German«).
   - Wählen Sie den einfachen Modus (»Easy Mode«).
   - Wählen Sie »Morphology«.
   - Klicken Sie auf »Run Tools«.

- Exportieren Sie die Analyse mit der Funktion »Download as Excel sheet«.
- Nutzen Sie ggf. auch die Informationen aus Kap. 2.3.
- Speichern Sie das Ergebnis unter dem Namen »Tagging_Flexionsmorphologie_getaggt_auto.xlsx«.

3. Sie können nun die beiden Analysen vergleichen, indem Sie die Spalte mit den Tagginginformationen aus der automatisch analysierten Datei neben die Spalte mit den flexionsmorphologischen Werten aus der manuell annotierten Datei kopieren (Sie werden möglicherweise an zwei Stellen alignieren – in Deckung bringen – müssen, weil die Tokenisierung des in WebLicht verwendeten Tokenisierers nicht mit der Tokenisierung in der manuell getaggten Datei übereinstimmt). Speichern Sie diese Vergleichsdatei unter dem Namen »Taggen_Flexionsmorphologie_getaggt_automatisch_manuell_Vergleich.xlsx«.

Lösungen

1. Lösungsdatei »Taggen_Flexionsmorphologie_getaggt_manuell.xlsx«: https://bit.ly/2OKivro
2. Lösungsdatei »Taggen_Flexionsmorphologie_getaggt_auto.xlsx«: https://bit.ly/2TS3oNl
3. Lösungsdatei »Taggen_Flexionsmorphologie_getaggt_automatisch_manuell_Vergleich.xlsx«: https://bit.ly/2UyecVu

Arbeitsaufgabe 2.2.6.7

Bearbeiten Sie die unter der Webadresse https://bit.ly/2TQsUY5 befindliche EXMARaLDA-Datei zur manuellen Analyse von Wortbildungsphänomenen.

- Ziel der Analyse ist es, jedem Token (Tokendefinition: Wort oder Satzzeichen) im Text einen Wert für den letzten stattgefundenen wortbildungsmorphologischen Prozess zuzuweisen. Flexion wird dabei nicht berücksichtig. Simplizia (einfache, wortbildungsmorphologisch nicht weiter zerlegbare Wortformen) werden als solche gekennzeichnet. Fremdwörter, deren morphologischer Status vor dem Hintergrund der deutschen Wortbildung intransparent ist, ebenso. Token, die keine Wörter (sondern Satzzeichen) sind, werden nicht weiter analysiert (sie erhalten das Tag »--«).
- Wenn Sie sich selber ein wortbildungsmorphologisches Tagset ausdenken (z. B. Komposition = "KOMP", Derivation = "DER" usw.) und dieses auf die Daten anwenden, werden Sie feststellen, dass bestimmte Wortformen schwer zuzuordnen sind und dass Sie häufig neue Kategorien hinzufügen müssen.
- Unter der Webadresse https://bit.ly/2Fe1Esk finden Sie eine Datei, die Sie mit EXMARaLDA einlesen können, so dass Ihnen wortbildungsmorphologische Werte vorgegeben werden. Gehen Sie nach dem Herunterladen der Datei wie folgt vor:
  - Klicken Sie in EXMARaLDA auf »View« > »Annotation panel«.
  - Klicken Sie auf dem Panel auf »Open«.
  - Lesen Sie die XML-Datei ein.

- Nun können Sie für jede aktive Zelle einen der vorgeschlagenen Werte einfüllen (Doppelklick auf die entsprechende Kategorie im Panel). Klicken Sie unten im Panel außerdem auf die Option »Auto jump«, um von Zelle zu Zelle springen.
- Speichern Sie die fertig bearbeitete Datei unter dem Namen »Tagging_Wortbildungsmorphologie_getaggt.exb«.

Lösung

Lösungsdatei »Tagging_Wortbildungsmorphologie_getaggt.exb«: https://bit.ly/2FTc16p

Arbeitsaufgabe 2.2.6.8

Überprüfen Sie die Eigennamenerkennung des TreeTaggers (der TagAnt-Version), indem Sie den Text unter der Webadresse https://bit.ly/2HAl1iT mit TagAnt verarbeiten (siehe die Hinweise zur Nutzung von TagAnt in Kap. 2.2.6.4).

- Konvertieren Sie die von TagAnt ausgegebene Datei »Eigennamenerkennung_tagged.txt« in das Format ANSI (nutzen Sie hierfür einen fortgeschrittenen Texteditor wie Notepad++, https://notepad-plus-plus.org/).
- Importieren Sie die Datei »Eigennamenerkennung_tagged.txt« in EXMARaLDA (über die Option »File« > »Import...« und die »Dateityp:«-Einstellung »Tree Tagger Output (*.txt)«.
- Fügen Sie eine Annotationsspur (Tier) »pos_korrigiert« hinzu (Option »Insert tier«) und übernehmen Sie die Werte der Spur »pos«.
- Korrigieren Sie die Analyse sämtlicher Eigennamen und derjenigen Token, die fälschlicherweise als Eigennamen erkannt wurden (das STTS-Tag für Eigennamen ist »NE«).
- Speichern Sie die Datei unter dem Namen »Eigennamenerkennung_TreeTagger_korrigiert.exb«.

Lösung

Lösungsdatei »Eigennamenerkennung_TreeTagger_korrigiert.exb«: https://bit.ly/2FNDohf

Hinweis: Zur Lösung gehört im Grunde nur die Ebene »pos_korrigiert«, in der Lösungsdatei sind aber weitere Ebenen angelegt, die zur Fehlerauswertung und Evaluation des Taggers dienen (siehe Kap. 2.5.3.1).

Arbeitsaufgabe 2.2.7.1

- Unter der Webadresse https://bit.ly/2Wee8ai finden Sie eine Version des bereits mehrfach verarbeiteten Beispieltextes mit annotierten Satzspannen (Ganzsätze). (Die Verarbeitung erfolgte mittels der o. g. Programmversion von Andreas Nolda.)
- Fügen Sie auf der noch unverarbeiteten Annotationsebene für Nebensätze allen untergeordneten Sätzen Werte für ihren syntaktischen Status zu. Hierzu müssen die entsprechenden Sätze zunächst durch Spannen gekennzeichnet und die Spannen anschließend mit Kürzeln (Tags) für den syntaktischen Status versehen (»gelabelt«) werden. Die möglichen Kategorien sind: Ergänzungssatz – Objektsatz (»SOBJ«); Ergänzungssatz – Subjektsatz (»SSUBJ«); Adverbialsatz (»SADV«); Attributsatz (»SATTR«).
  Unter der Webadresse https://bit.ly/2OhE4iv können Sie eine XML-Datei beziehen, die Sie in das Annotationspanel von EXMARaLDA einlesen können. Beachten Sie ggf. die Hinweise in Aufgabe 1 von Kap. 2.2.6.7 zur Konfiguration des Annotationspanels.
- Speichern Sie die fertig analysierte Datei unter dem Dateinamen »Satzspannen_Nebensatz_annotiert.exb«.

Lösung

Lösungsdatei »Satzspannen_Nebensatz_annotiert.exb«: https://bit.ly/2FQwGHh

Arbeitsaufgabe 2.2.7.2

- Laden Sie die unter der Webadresse https://bit.ly/2TWCOXB verfügbare Datei herunter und öffnen Sie diese in EXMARaLDA (Doppelklick bzw. »File« > »Open...« im Partitureditor).
- Annotieren Sie die Datei, indem Sie den Sätzen auf der Annotationsebene »Topologische Felder« die Kategorien »VF« für »Vorfeld«, »LSK« für »linke Satzklammer«, »MF« für »Mittelfeld«, »RSK« für »rechte Satzklammer« und »NF« für »Nachfeld« zuweisen.
  - Orientieren Sie sich bei der Vergabe der Kategorien an der Übersicht in Tab. 2.3 und ziehen Sie Sie ggf. die in diesem Kapitel referenzierte Literatur zu Rate.
  - Unter der Webadresse https://bit.ly/2JvfzQo können Sie eine XML-Datei beziehen, die Sie in das Annotationspanel von EXMARaLDA einlesen können. Beachten Sie ggf. die Hinweise in Aufgabe 1 von Kap. 2.2.6.7 zur Konfiguration des Annotationspanels.
- Speichern Sie die bearbeitete Datei unter dem Namen »Topologische_Felder_annotiert.exb«.

Lösung

Lösungsdatei »Topologische_Felder_annotiert.exb«: https://bit.ly/2K3kEQ5

Arbeitsaufgabe 2.2.7.3

- Laden Sie die unter der Webadresse https://bit.ly/2TVPrlG verfügbare Datei herunter und öffnen Sie diese in EXMARaLDA (Doppelklick bzw. »File« > »Open...« im Partitureditor).
- Annotieren Sie die Datei, indem Sie zunächst Spannen für die satzgliedwertigen Phrasen und verbalen Cluster ziehen und anschließend Phrasenkategorien hinzufügen. Halten Sie sich bei der Phrasenanalyse an die o. g. Hinweise zur Analyse von Maximalphrasen.
  Nutzen Sie für das Auszeichnen (Labeln) die unter der Webadresse https://bit.ly/2FrAfVm beziehbare XML-Datei für das EXMARaLDA-Annotationspanel: Aktivieren Sie die Funktion »Auto jump« und nutzen Sie im Fall von nicht auszufüllenden Zellen (z. B. Satzzeichen) die Option »WEITERKLICKEN«.
- Speichern Sie die bearbeitete Datei unter dem Namen »Phrasenchunking_annotiert.exb«.

Lösung

Lösungsdatei »Phrasenchunking_annotiert.exb«: https://bit.ly/2UrRtdj

Arbeitsaufgaben 2.2.7.5

Mit den folgenden Teilaufgaben können Sie die Anreicherung von unverarbeiteten Textdaten mit Dependenzparses nachvollziehen.

1. Verarbeiten Sie den unter der Webadresse https://bit.ly/2CvGbum erhältlichen Text mit dem auf das Deutsche ausgelegte Parser ParZu (https://github.com/rsennrich/parzu), indem Sie die online befindliche Demoversion verwenden: https://pub.cl.uzh.ch/demo/parzu/.
   - Kopieren Sie hierzu den Text in das Input-Fenster, wählen Sie das Ausgabeformat »CoNLL« und klicken Sie »SEND«.
   - Kopieren Sie anschließend die erzeugten Daten in eine Textdatei, die Sie unter dem Dateinamen »Parsen.conll« abspeichern. Achten Sie darauf, dass die Datei in der Kodierung »UTF-8« gespeichert wird.

Mit der folgenden Aufgabe können Sie die geparsten Daten visualisieren.

2. Lesen Sie die Datei in das Programm »DG Annotator« (https://bit.ly/2FnDoEx) ein (nach der Installation des Programms muss man die Datei »dga.jar« ausführen; eine aktuelle Java-Installation ist erforderlich: https://java.com/de/download/).
   - Starten Sie nun das Programm und wählen Sie die Einstellung »Configure« > »Corpus...« > »German«.
   - Laden Sie nun die Datei »Parsen.conll«. Sie können diese auch unter der Webadresse https://bit.ly/2TNFzuQ beziehen. Hinweis: Wenn Sie im »Öffnen«-Dialog des DGA-Annotators den Dateityp »XML, CoNLL file« auswählen, werden nur die verfügbaren Dateien mit der relevanten Dateiendung angezeigt.
   - Sie können nun die analysierten Sätze einzeln betrachten.

Zum Visualisieren und Korrigieren der Daten in Kim Gerdes' ›Arborator‹ s. Aufgabe 3.

3. Öffnen Sie die Webseite https://arborator.ilpga.fr/q.cgi in Google Chrome.
   - Kopieren Sie die Daten aus der Datei »Parsen.conll« in das große Fenster rechts (löschen Sie vorher die dort befindlichen Beispieldaten).
   - Sie können sich die Parses nun anschauen. Um sie mit dem passenden Tagset für syntaktische Funktionen und Wortarten bearbeiten zu können, beziehen Sie die Tags aus der Datei, die Sie unter der Webadresse https://bit.ly/2Yc54oh erhalten. Kopieren Sie die Funktionstags in die Box »additional functions« in Arborator und kopieren Sie die STTS-Wortartentags in die Box »additional POS tags«.
   - Nun können Sie die Daten mit den für sie vorgesehenen Kategorien editieren. Die automatisch erzeugten Parses sind so akkurat, dass es nur wenig zu korrigieren gibt (ein Präpositionalobjekt wurde nicht erkannt, die Anbindung des *dass*-Satzes ist nicht korrekt und ggf. gehört *schon* zu *lange*).
   - Kopieren Sie das Ergebnis der Korrektur (die Tabellendaten auf der rechten Seite) in eine Textdatei namens »Dep_Parse_korrigiert.txt«.

Lösungen

1. Lösungsdatei »Parsen.conll«: https://bit.ly/2VhcJjc
2. Lösungsdatei »Dep_Parse_korrigiert.txt«: https://bit.ly/2FP8rJm

Arbeitsaufgabe 2.2.8

- Öffnen Sie in einem Internetbrowser die Seite https://pub.cl.uzh.ch/demo/corzu/ (es handelt sich um die Online-Demoversion des Programms CorZu.
- Geben Sie die Sätze *Erwin schreibt Klara einen ausführlichen Brief. In diesem kann er ihr alles am besten erklären.* als Text in das Eingabefeld ein.
- Betätigen Sie die Einstellung »CoNLL« und bestätigen Sie (»Proceed«).
- Überführen Sie die ausgegebenen Daten nach Arborator (https://arborator.ilpga.fr/q.cgi; siehe Aufgabe 1 in Kap. 2.2.7.5 für Bedienungshinweise).
- Fügen Sie die fehlende Koreferenzbeziehung (3) hinzu (dies müssen Sie auf der rechten Seite in den Tabellendaten tun) und korrigieren Sie die Dependenzannotationen.
- Speichern Sie das Ergebnis der Bearbeitung (die Tabelleninformationen rechts) in einer Textdatei namens »Koref_korrigiert.txt«.

Lösung

Lösungsdatei »Koref_korrigiert.txt«: https://bit.ly/2VbsurY

Arbeitsaufgabe 2.2.9

- Führen Sie die Analyse noch einmal durch, so dass der zweite Satz (mit dem dritten Satz zusammen) Satelliten für den ersten Satz mit der Funktion »Elaboration« sind.
- Speichern Sie das Ergebnis als »RST_Loesung2.rs3«.

Lösung

Lösungsdatei »RST_Loesung2.rs3«: https://bit.ly/2TU13S2

Arbeitsaufgaben 2.4.7

1. Unter der Webadresse https://bit.ly/2OltHuo erhalten Sie eine Audiodatei, die einen dreizehn Sekunden langen Ausschnitt eines Dialogs zwischen zwei Sprecherinnen sowie ein dieser Datei zugeordnetes EXMARaLDA-Transkript enthält. Das Gespräch wird in voller Länge auf der Webseite http://agd.ids-mannheim.de/gat.shtml angeboten; eine Transkription dieses Gesprächs im Sinne des GAT-Basis- und auch des Feintranskripts finden Sie in Selting et al. (2009), S. 42 f. ebenso auf der Webseite. Für die Bearbeitung der folgenden Aufgaben können Sie sich an dem Beispiel des Basistranskripts orientieren; der Übungseffekt ist jedoch stärker, wenn Sie die Umsetzung der Aufgaben zunächst ohne Vorlage versuchen.
   - Entpacken Sie die zwei Dateien in denselben Ordner.
   - Öffnen Sie die EXMARaLDA-Datei.
   - Spielen Sie die Audiodatei in einem separaten Audioplayer vollständig und wiederholt ab.
   - Beginnen Sie mit dem Anfang des Gesprächs: Schreiben Sie die ca. ersten fünf Sekunden des Gesprochenen in literarischer Umschrift in die erste Zelle von Sprecherin 1 in den EXMARaLDA-Partitureditor. Transkribieren Sie bis zu dem Wort *lassen* (erstes Vorkommen). Achten Sie auf kontinuierliche Kleinschreibung und die Kennzeichnung bestimmter Aussprachevarianten durch die Abweichung von der Schreibnorm.
   - Übernehmen Sie das von Sprecherin 2 Gesprochene in die erste Zelle. Sie können die Überlappungen durch die in GAT 2 vorgesehenen eckigen Klammern (auf beiden Ebenen) kennzeichnen oder zunächst den spezifischen Überlappungsbereich unmarkiert lassen.
   - Gehen Sie gemäß dem bisherigen Vorgehen bis zum Ende der Audiodatei vor. Die Entertaste erzeugt ein neues Event rechts von dem aktivierten Event. Die Tabstopptaste springt zum nächsten Event, sofern es bereits vorhanden ist.
2. Laden Sie unter der Webadresse https://bit.ly/2FndGQr eine EXMARaLDA- und eine FOLKER-Datei herunter, die mit der Audiodatei im ersten Download oben verknüpft sind. Im folgenden Bearbeitungsschritt geht es um die Alignierung des transkribierten Texts mit den passenden Ausschnitten aus der Audiodatei.
   Da EXMARaLDA und FOLKER so konzipiert sind, dass Audiosequenzen beim Transkribieren zugewiesen werden, müssen Sie den in Aufgabe 1

transkribierten Text noch einmal eingeben, während Sie Abschnitte in der Audiodatei zuweisen. Sie können aber die bereits transkribierte Datei separat öffnen und die Zelleninhalte in die neue Bearbeitung hinüberkopieren.

Gehen Sie bei der Bearbeitung folgendermaßen vor:

a) Bearbeitung in EXMARaLDA
- Öffnen Sie die EXMARaLDA-Datei (»Transkript_v1_Audio.exb«).
- Klicken Sie im Editor den Knopf »Append interval«. Sie hören die ersten zwei Sekunden der Aufnahme. Schreiben Sie den entsprechenden Text in die erste Zelle von Sprecherin 1. (Der ausgewählte Ausschnitt entspricht genau der ersten Intonationsphrase im GAT-2-Beispieltranskript. Wenn Sie die erste Einheit verlängern wollen, können Sie dies durch eine Vergrößerung des Audioausschnitts tun.)
- Klicken Sie wieder »Append interval«. Sie hören die nächsten zwei Sekunden der Aufnahme und müssen den Audioausschnitt verlängern (bis ungefähr 4,7 Sekunden in der Aufnahme). Geben Sie den entsprechenden Text ein.
- Fügen Sie bei Sprecherin 2 die Redeanteile hinzu. Achten Sie darauf, dass die ersten zwei Interjektionen (»ja« und »hm«) mit einer Intonationsphrase von Sprecherin 1 überlappen (Sie können diese in zwei Events bzw. Zellen unterteilen) und dass die dritte Interjektion (»hm«) nicht überlappt.
- Führen Sie die Prozedur fort, bis die Aufnahme vollständig bearbeitet ist.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v1_Minimaltranskript.exb«.

b) Bearbeitung in FOLKER
- Öffnen Sie die Datei »Transkript_v1_Audio.flk« mit FOLKER (FOLKER gehört zu den Programmen, die man mit EXMARaLDA auf der Webseite www.exmaralda.org herunterladen kann).
- Der erste Transkriptionsausschnitt ist bereits eingerichtet: Klicken Sie auf eine Spalte der ersten angelegten Zeile. Sie können nun den ersten Audioausschnitt abspielen. Geben Sie den entsprechenden Text in die Spalte »Transkriptionstext« ein.
- Klicken Sie ganz rechts auf »Append new segment« (blaues Plus-Symbol mit grünem Pfeil). Hierdurch fügen Sie der Transkription einen neuen Audioausschnitt mit einem neuen Transkriptionssegment hinzu. Hören Sie den hinzugefügten Ausschnitt, erweitern Sie ihn bis zur nächsten Segmentgrenze und geben Sie den Transkriptionstext in die entsprechende Zeile und Spalte ein.
- Klicken Sie ganz rechts auf »New segment« (blaues Plus-Symbol). Hierdurch fügen Sie demselben Ausschnitt eine neue Transkriptionszeile hinzu. Wählen Sie in der Spalte »Speaker« »Sprecherin 2« und fügen Sie der Zeile das von Sprecherin 2 Gesprochene hinzu (übernehmen Sie beide Interjektionen (»ja« und »hm« in dieselbe Transkriptionszeile).
- Fahren Sie mit der Prozedur fort, bis die Aufnahme vollständig bearbeitet ist. Achten Sie auf die korrekte Zuweisung der Sprecherinnen zu den jeweiligen Beiträgen.

- Speichern Sie das Ergebnis unter dem Namen »Transkript_v1_Minimaltranskript.flk«.
- Sie können die gespeicherte Datei mit der Funktion »Import« in EXMARaLDA öffnen und dort weiterbearbeiten (FOLKER ist für das Transkribieren konzipiert, EXMARaLDA für das Transkribieren und weitere Annotieren). Die in EXMARaLDA importierte FOLKER-Transkription sollte identisch mit der in EXMARaLDA erstellten Transkription sein. Ausnahme: Die überlappenden Interjektionen von Sprecherin 2 können in der ursprünglichen EXMARaLDA-Transkription in einzelnen Zellen stehen (Sie können in der von FOLKER nach EXMARaLDA konvertierten Datei die Zelle aufsplitten).
- In den folgenden Aufgaben werden dem bislang erstellten Minimaltranskript weitere bereits besprochene sprachliche Merkmale hinzugefügt und es werden hierfür jeweils eigene Annotationsebenen angelegt.

3. Bauen Sie auf der erstellten Fassung des Minimaltranskripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2JwEcfp). Fügen Sie die Analyse von Akzentsilben hinzu, indem Sie diese mit Majuskeln (Großbuchstaben) markieren. Gehen Sie dabei wie folgt vor:

- Fügen Sie pro Sprecherin je eine Annotationsspur »Akzent« hinzu (bei markierter Annotationszeile STRG-i für »Insert tier« eingeben, Sprecherin zuweisen, Typ »Annotation« wählen, Namen der Spur eingeben, die Inhalte der entsprechenden Transkriptionsspur in die neue Spur kopieren). Ordnen Sie die Annotationsspuren gemäß Abb. 2.20 an.

Sprecherin_1 [Umschrift]	ja die vierziger generation	so das wahnsinnig viele
Sprecherin_1 [Akzent]	ja die vierziger generation	so das WAHNsinnig vie
Sprecherin_2 [Umschrift]		ja
Sprecherin_2 [Akzent]		ja

Abb. 2.20: Anordnung der Annotationsspuren in EXMARaLDA

- Markieren Sie in der hinzugefügten Spur die Akzentsilben mit Majuskeln. Achten Sie dabei auf die Markierung genau der Silbeneinheit.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v2_Akzent.exb«.

4. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2CqbW8n). Analysieren Sie zusätzlich Pausen. Gehen Sie dabei analog zur Prozedur in Aufgabe 3 vor: Fügen Sie den Spuren »Umschrift« und »Akzent« pro Sprecherin jeweils eine Spur »Pausen« hinzu (achten Sie auf die korrekten Einstellungen für die Sprecherzuordnung und den Informationstyp).

- Entnehmen Sie die Pausentypen dem Dokument Selting et al. (2009), S. 21 f. bzw. den Informationen oben im Kapitel.
- Fügen Sie die Pausenmarkierungen auf derjenigen Spur derjenigen Sprecherin hinzu, die gerade den Turn besitzt. Trennen Sie die Pausenmarkierungen mit Leerzeichen von Text ab.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v3_Pausen.exb«.

5. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2JsFSXl). Fügen Sie die Analyse von Tonhöhenverläufen am Ende der Transkriptionssegmente hinzu, indem Sie eine Spur namens »Tonhöhe« hinzufügen und diese mit den Kategorien aus dem Dokument Selting et al. (2009), S. 21 f. versehen.
   - Schreiben Sie die Zeichen ohne Leerzeichenabstand an das letzte Wort einer jeden Zelle. Berücksichtigen Sie also jede als abgeschlossene Intonationsphrase interpretierte Einheit mit einem Annotationswert am Ende der Phrase.
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v4_Tonhoehe.exb«.
6. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2Wgc69U). Fügen Sie eine Analyse der Artikulationslänge von Vokalen und Konsonanten hinzu, indem Sie eine Spur namens »Längung« hinzufügen und diese mit den Kategorien aus dem Dokument Selting et al. (2009), S. 21 f. versehen.
   - Hören Sie durch die Audiodatei und überlegen Sie, welche Laute eine relative Länge gegenüber ihrer Normallautung besitzen. Sie werden feststellen, dass dies relativ schwer zu bemessen ist.
   - Lösungshinweis: Markieren Sie ganz zu Beginn der Aufnahme einen Vokal und in der zweiten Intonationsphrase (nach Selting et al. 2009) einen Konsonanten und einen Vokal, jeweils mit einer einfachen Längung (einfacher Doppelpunkt).
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v5_Laengungen.exb«.
7. Bauen Sie auf der erstellten Fassung des Transkripts auf (zum Vergleich mit einer Lösung: https://bit.ly/2UNU0vt). Fügen Sie der Gesamtanalyse schnelle Anschlüsse hinzu, indem Sie eine Spur namens »Anschlüsse« hinzufügen und schnelle Anschlüsse von Intonationsphrasen bei derselben Sprecherin analysieren:
   - Markieren Sie jeweils am Ende der vorangehenden Zelle und am Beginn der nachfolgenden mit einem Gleichheitszeichen (ohne Leerzeichen zu den umliegenden Zeichen) den unmittelbaren Anschluss.
   - Speichern Sie das Ergebnis unter dem Namen »Transkript_v6.exb«.

Lösungen

1. a) Lösungsdatei »Transkript_v1_Minimaltranskript.exb«: https://bit.ly/2YEMwgC
   b) Lösungsdatei »Transkript_v1_Minimaltranskript.flk«: https://bit.ly/2TUQiz7
2. Lösungsdatei »Transkript_v2_Akzent.exb«: https://bit.ly/2FXxziF
3. Lösungsdatei »Transkript_v3_Pausen.exb«: https://bit.ly/2JZdQCT
4. Lösungsdatei »Transkript_v4_Tonhoehe.exb«: https://bit.ly/2YMjs6R
5. Lösungsdatei »Transkript_v5_Laengungen.exb«: https://bit.ly/2UsTGW4
6. Lösungsdatei »Transkript_v6.exb«: https://bit.ly/2WLA5hr

Arbeitsaufgabe 2.4.8

Löschen Sie aus jeder Zelle der EXMARaLDA-Datei »Transkript_v6.exb« (https://bit.ly/2WkY7jd) sämtliche für das jeweilige Phänomen irrelevante Informationen:

- Behalten Sie auf der Ebene »Akzent« nur die Wörter mit Fokusakzent.
- Behalten Sie auf der Ebene »Pausen« nur die Pauseninformationen.
- Behalten Sie auf der Ebene »Tonhöhe« nur die Zeichen für die Intonationskontur am Intonationsphrasenende.
- Behalten Sie auf der Ebene »Längung« nur die Wörter mit Längungen.
- Behalten Sie auf der Ebene »Anschluss« nur die GAT-Segmente mit Anschluss zu einem Nachbarsegment (der Text kann hier beibehalten werden).
- Sie können durch eine geschickte Segmentierung des Gesamttranskripts versuchen, die einzelnen Phänomene möglichst gut mit ihrer Position auf der Ebene des Minimaltranskripts (»Umschrift«) zu verknüpfen. Wortweise lässt sich dies jedoch erst nach einer Tokenisierung der Daten gemäß einer wortbezogenen Tokendefinition erreichen (siehe Kap. 2.4.10).
- Speichern Sie die bearbeitete Datei unter dem Namen »Transkript_v7.exb«.

Lösung

Lösungsdatei »Transkript_v7.exb« https://bit.ly/2WAqqds

Arbeitsaufgabe 2.4.10

Arbeiten Sie mit der letzten Version des komplexen Transkripts weiter. Die Lösung zur letzten Arbeitsaufgabe in Kap. 2.4.8 erhalten Sie unter der Webadresse https://bit.ly/2Y84DLP. Führen Sie in diesem Dokument die folgenden Arbeitsschritte zur Normalisierung der Daten durch.

- Fügen Sie für die beiden Sprecherinnen eine Normalisierungsebene mit der Bezeichnung »norm« hinzu, wobei Sie den Inhalt der jeweiligen Ebene »Umschrift« kopieren (siehe ggf. die Hinweise im Aufgabenteil von Kap. 2.4.7).
- Passen Sie den Inhalt der Transkription an die deutsche Standardorthographie inklusive Zeichensetzung an.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v8.exb« ab.
- Bearbeiten Sie auch Aufgabe 1 im nachfolgenden Kap.2.4.11 , um die Weiterverarbeitung der normalisierten Ebene zu behandeln.

Lösung

Lösungsdatei »Transkript_v8.exb«: https://bit.ly/2TQrEzp

Arbeitsaufgabe 2.4.11

Bei der Fertigstellung dieses Buchs existierte noch keine einfache Möglichkeit, beliebige Spuren in EXMARaLDA oder einem der mit EXMARaLDA kompatiblen Programme zu tokenisieren und zu taggen. Andreas Nolda erstellt derzeit eine Anwendung für die EXMARaLDA-Programmversion ›EXMARaLDA (Dulko)‹ (https://andreas.nolda.org/software.html), mit der verschiedene automatisierte Transformationen, u. a. der Taggingprozess mit dem TreeTagger für beliebige Spuren ermöglicht werden. Um nachzuvollziehen, zu welchen Ergebnissen eine Anreicherung von Korpusdaten wie z. B. der Datei »Transkript_v8.exb« (https://bit.ly/2TmwSCI) führen kann, befolgen Sie die folgenden Schritte:

- Laden Sie die Datei »Transkript _v8.exb« unter dem angegebenen Weblink herunter. Sie enthält eine zur anschließend getaggten Datei passende Aufteilung der EXMARaLDA-Grundsegmente (Timeline-Items).
- Laden Sie eine zweite EXMARaLDA-Datei herunter, in welcher die beiden »norm«-Spuren der Datei »Transkript_v8.exb« mithilfe von EXMARaLDA (Dulko) getaggt wurden: https://bit.ly/2ugYcs6.
- Verschmelzen Sie die beiden Dateien, indem Sie »Transkript_v8.exb« in EXMARaLDA öffnen und die andere Datei mithilfe der Funktion »Transcription« > »Merge transcriptions...« hinzufügen.
- Ordnen Sie die unten hinzugefügten Annotationsebenen den jeweiligen Sprecherinnen zu, indem Sie sie nach oben verschieben (Funktion »Change tier order...« oben im grafischen Menü) und analog zueinander anordnen.
- Speichern Sie das Ergebnis unter dem Namen »Transkript_v9.exb«.

Lösung

Lösungsdatei »Transkript_v9.exb«: https://bit.ly/2VdvzaZ

Arbeitsaufgabe 2.5.2.1

- Laden Sie die Datei »Uebereinstimmung_prozentual.xlsx« von der Webadresse https://bit.ly/2FuSRTl herunter. Öffnen Sie die Datei in LibreOffice (oder OpenOffice) Calc oder Microsoft Excel und durchlaufen Sie die folgenden Schritte:
- Klicken Sie in die Zelle E2 und doppelklicken Sie dann auf das Symbol rechts unten in der Zelle. Sie bekommen sämtliche Unterschiede zwischen einer automatisch generierten Wortartenanalyse des TreeTaggers und der manuellen Korrektur angezeigt.
- Berechnen Sie anhand der Gesamtzahl der pos-Vergaben und der Übereinstimmungen die prozentuale Korrektheit des TreeTaggers bzw. die Übereinstimmung in Prozent.

Lösung

- Anzahl vergebener Tags: 52
- Anzahl Fehler: 4 → Anzahl Übereinstimmungen: 48
- Prozentuale Übereinstimmung: 48/52≈0,92 (also gerundet 92 %)

Arbeitsaufgabe 2.5.3.1

- Laden Sie das Ergebnis der Eigennamenkorrektur (siehe Kap. 2.2.6.8; die Datei »Eigennamenerkennung_TreeTagger_korrigiert.exb«) in EXMARaLDA. Sie erhalten die Datei auch unter der Webadresse https://bit.ly/2FnX7UB.
- Ermitteln Sie anhand der Annotationsebenen »pos« (das ist die automatische Analyse des TreeTaggers, mithilfe von TagAnt erzeugt) und »pos_korrigiert« (der manuellen Korrektur) die Werte von
  a) Precision,
  b) Recall,
  c) F-score
  für die Erkennung von Eigennamen durch den TreeTagger.

Lösung

Der Berechnung liegen folgende Daten zugrunde:

- Anzahl von Eigennamen (NE) im Text: 10 (Sie erhalten diese Zahl, indem Sie die Ebene »pos_korrigiert« nach dem Wert »NE« durchsuchen.)
- Anzahl vom Tagger analysierter Eigennamen: 11 (Sie erhalten diese Zahl, indem Sie die Ebene »pos« nach dem Wert »NE« durchsuchen.)
- Anzahl korrekter Zuweisung: 9 (Sie erhalten diese Zahl, indem Sie die Ebene »NE_korrekte_Zuweisung« nach dem Wert »+« durchsuchen.)
  (Anzahl nicht erkannter Eigennamen: 1 (Diese Anzahl ergibt sich aus der Differenz der Anzahl im Text enthaltenen Eigennamen und der Anzahl korrekt zugewiesener Eigennamen. Man kann auch die Anzahl des Werts »-« auf der Ebene »NE_nicht_erkannt« ermitteln.))
  (Anzahl falscher Zuweisungen: 2 (Diese Zahl ergibt sich aus der Differenz der von Tagger vergebenen Eigennamen und der Anzahl der korrekten Zuweisungen. Sie erhalten diese Zahl auch, indem Sie die Ebene »NE_falsche_Zuweisung« nach dem Wert »-« durchsuchen.))
  a) Precision: $9/11 \approx 0{,}82$
  b) Recall: $9/10 = 0{,}9$
  c) F-score:

$$F = \frac{2\ (0{,}82 \times 0{,}9)}{0{,}82 + 0{,}9} \approx 0{,}86$$

Arbeitsaufgaben 3.1.2.2

1. Formulieren Sie Suchausdrücke für die Suche nach der Imperativform des Verbs *halten* in allen drei Anfragesprachen und testen Sie diese Anfragen auf den in Kap. 3.1.2 vorgestellten Suchportalen.
   - Was ist das Problem an der ausgegebenen Treffermenge?
   - Welche Informationen müsste man abfragen, um dieses Problem zu vermeiden?
2. Sie wollen Parenthesen (Einschübe im Satz oder herausgestellte Nachträge) finden. Mit welchen Suchanfragen für die jeweiligen Suchsysteme können Sie dies erreichen?

Lösungen

1. ANNIS-Anfrage:

```
tok="halt"
```

CQP-Anfrage:

```
[word="halt"]
```

TIGERSearch-Anfrage:

```
[word="halt"]
```

Es werden nicht nur Imperative von *halten* ausgegeben, sondern auch Vorkommen der Modalpartikel *halt* und andere nicht gewollte Vorkommen der Wortform (z. B. mit apokopiertem Schwa). Großgeschriebene Vorkommen am Satzanfang wiederum werden nicht gefunden. Letzterem kann man entgegenwirken, wenn man zusätzlich nach der Wortform *Halt* sucht, dann werden allerdings auch Vorkommen des Nomens *Halt* gefunden. Man müsste also Wortarten kontrollieren können, um ungewollte Treffer zu vermeiden.

2. ANNIS-Anfrage:

```
tok="-"
```

CQP-Anfrage:

```
[word="-"]
```

TIGERSearch-Anfrage:

```
[word="-"]
```

Hinweis für alle drei Suchanfragen: Es macht einen Unterschied, ob in den Suchsystemen ein kurzer oder langer Bindestrich eingegeben wird.

Arbeitsaufgabe 3.1.2.3

1. Formulieren Sie Suchen nach den folgenden Wortformmustern für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.
   a) Wörter, die mit großgeschriebenem *Z* beginnen.
   b) Wörter, die an irgendeiner Stelle den Bestandteil *-bar-* enthalten.
   c) Wörter, die an irgendeiner Stelle zwei *e* oder mehr enthalten.

Lösung

a) ANNIS-Anfrage:

```
tok=/Z.*/
```

CQP-Anfrage:

```
[word="Z.*"]
```

TIGERSearch-Anfrage:

```
[word="Z.*"]
```

b) ANNIS-Anfrage:

```
tok=/.*bar.*/
```

CQP-Anfrage:

```
[word=".*bar.*"]
```

TIGERSearch-Anfrage:

```
[word=".*bar.*"]
```

c) ANNIS-Anfrage:
`tok=/.*ee+.*/`
CQP-Anfrage:
`[word=".*ee+.*"]`
TIGERSearch-Anfrage:
`[word=".*ee+.*"]`

Arbeitsaufgabe 3.1.2.4

Hinweis zu der folgenden Aufgabe: Beachten Sie, dass es sich um fingierte Fälle handelt. An dieser Stelle können Sie deshalb nicht wie sonst die formulierten Suchanfragen an authentischen Beispielen ausprobieren.

Stellen Sie sich vor, Sie haben Zugriff auf ein Korpus mit einer Annotationsebene »Exklamation« (Kürzel: EK). Auf dieser Ebene sind exklamative Interjektionen (Kürzel: ITJ), vokativ gebrauchte Nomina (Kürzel: VOK) und imperative Verben (Kürzel: IMP) annotiert. Die Ebene für die fortlaufenden Wortformen bzw. den tokenisierten Text ist mit dem Kürzel TXT (für »Text«) bezeichnet.

a) Wie lautet die Suchanfrage, die in diesem Korpus alle imperativ gebrauchte Verben findet?
b) Wie finden Sie in dem Korpus die Wortform »Mist«?

Lösung

a) `EK="IMP"`
b) `TXT="Mist"`

Arbeitsaufgabe 3.1.2.5

Formulieren Sie Suchen nach den folgenden Lemmata und Lemma-Mustern für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle Wortformen von Wörtern, deren Grundform auf *-lich* endet.
b) Finden Sie alle Wortformen des Flexionsparadigmas, zu dem die Form *wären* gehört.

Hinweis: Sollte eine bestimmte Form eines Paradigmas (wie *wäre*) durch die entsprechende Lemmasuche nicht gefunden werden, so heißt dies nicht, dass die Suche nicht funktioniert, sondern lediglich, dass die entsprechende Form im durchsuchten Korpus nicht enthalten ist.

Lösung

a) ANNIS-Anfrage:
`lemma=/.*lich/`
CQP-Anfrage:
`[lemma=".*lich"]`
TIGERSearch-Anfrage:
`[lemma=".*lich"]`

b) ANNIS-Anfrage:
`lemma=/sein/`
CQP-Anfrage:
`[lemma="sein"]`
TIGERSearch-Anfrage:
`[lemma=/sein/]`

Arbeitsaufgabe 3.1.2.6

Formulieren Sie Suchen nach den folgenden Wortarten gemäß dem STTS-Tagset (siehe Kap. 2.2.6.2) für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle subordinierenden (unterordnenden) Konjunktionen (bzw. Subjunktionen) im Korpus.
b) Finden Sie alle finiten Verben im Korpus.
c) Finden Sie alle Nomina, inklusive Eigennamen, im Korpus.

Lösung

a) ANNIS-Anfrage:
`pos="KOUS"`
CQP-Anfrage:
`[pos="KOUS"]`
TIGERSearch-Anfrage:
`[pos="KOUS"]`
b) ANNIS-Anfrage:
`pos=/V.FIN/`
CQP-Anfrage:
`[pos="V.FIN"]`
TIGERSearch-Anfrage:
`[pos=/V.FIN/]`
c) ANNIS-Anfrage:
`pos=/N./`
CQP-Anfrage:
`[pos="N."]`
TIGERSearch-Anfrage:
`[pos=/N./]`

Arbeitsaufgabe 3.1.2.7

Formulieren Sie Suchen nach den folgenden Flexionskategorien für die einzelnen Suchsysteme. Wählen Sie dazu jeweils das TIGER-Korpus in der entsprechenden Instanz. Die Richtlinien zur Vergabe der flexionsmorphologischen Tags sind in Crysmann et al. 2005 (https://bit.ly/2Hwj9rL) formuliert. Vergleichen Sie auch das Kap. 2.2.6.6 zur Annotation von Flexionsmorphologie. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Wörter im Genitiv.
b) Finden Sie Pronomina und Verben in der ersten Person. (Da die erste Person genau bei Pronomina und Verben annotiert ist, müssen Sie die Suche nicht auf die entsprechenden Wortarten einschränken.)
c) Finden Sie Verben in Präteritalform. (Da die Vergangenheitsstammform ausschließlich auf Verben zutrifft, müssen Sie die Suche nicht auf Verben einschränken.)

Lösung

a) ANNIS-Anfrage:
```
morph=/.*Gen.*/
```
CQP-Anfrage:
```
[morph=".*Gen.*"]
```
TIGERSearch-Anfrage:
```
[morph=/.*Gen.*/]
```
b) ANNIS-Anfrage:
```
morph=/1.*/
```
CQP-Anfrage:
```
[morph="1.*"]
```
TIGERSearch-Anfrage:
```
[morph=/1.*/]
```
c) ANNIS-Anfrage:
```
morph=/.*Past.*/
```
CQP-Anfrage:
```
[morph=".*Past.*"]
```
TIGERSearch-Anfrage:
```
[morph=/.*Past.*/]
```

Arbeitsaufgabe 3.1.2.8

Formulieren Sie Suchen nach den folgenden Alternativen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie durch die Verkettung von Alternativen alle umgelauteten Formen des Verbs *haben* (z. B. *hätten*).
b) Finden Sie durch die Verkettung von Alternativen die STTS-Wortarten Postposition und/oder Zirkumposition.
c) Finden Sie durch die Verkettung von Alternativen die (pronominale) Form *sie* auch am Satzanfang.
d) Finden Sie durch die Verkettung von Alternativen Wortformen, die auf einem Nasal enden.

Lösung

a) ANNIS-Anfrage:
```
tok=/hätt(e|en|et|est)/
```
CQP-Anfrage:
```
[word="hätt(e|en|et|est)"]
```
TIGERSearch-Anfrage:
```
[word=/hätt(e|en|et|est)/]
```

b) ANNIS-Anfrage:
```
pos=/APP(O|RART)/
```
CQP-Anfrage:
```
[pos="APP(O|RART)"]
```
TIGERSearch-Anfrage:
```
[pos=/APP(O|RART)/]
```
c) ANNIS-Anfrage:
```
tok=/(S|s)ie/
```
CQP-Anfrage:
```
[word="(S|s)ie"]
```
TIGERSearch-Anfrage:
```
[word=/(S|s)ie/]
```
d) ANNIS-Anfrage:
```
tok=/.*(n|ng|m)/
```
CQP-Anfrage:
```
[word=".*(n|ng|m)"]
```
TIGERSearch-Anfrage:
```
[word=/.*(n|ng|m)/]
```

Arbeitsaufgabe 3.1.2.9

Formulieren Sie Suchen mit Zeichenmengen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie alle Lemmata, die mit einem Umlaut beginnen.
b) Finden Sie alle Wortformen, die mit einem kleingeschriebenen Vokal beginnen.
c) Finden Sie Wörter, die nicht auf einem Plosiv anlauten.

Lösung

a) ANNIS-Anfrage:
```
lemma=/[öäüÖÄÜ].*/
```
CQP-Anfrage:
```
[lemma="[öäüÖÄÜ].*"]
```
TIGERSearch-Anfrage:
```
[lemma=/[öäüÖÄÜ].*/]
```
b) ANNIS-Anfrage:
```
tok=/[euioayöäü].*/
```
CQP-Anfrage:
```
[word="[euioayöäü].*"]
```
TIGERSearch-Anfrage:
```
[word=/[euioayöäü].*/]
```
c) ANNIS-Anfrage:
```
tok=/[^tpdgkbqz].*/
```
CQP-Anfrage:
```
[word="[^tpdgkbqz].*"]
```

TIGERSearch-Anfrage:
`[word=/[^tpdgkbqz].*/]`

Hinweis zu allen drei Suchanfragen: Die Kategorie »Plosiv« lässt sich graphematisch auch anders definieren, z. B. ohne die Grapheme *q* und *z*.

Arbeitsaufgabe 3.1.2.10

Formulieren Sie Suchen mit bestimmten Groß- und Kleinschreibungsanforderungen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Vorkommen der Interjektion *oh* mit beliebig vielen Abfolgen von *o* und *h* sowie möglichst vielen verschiedenen Varianten von Groß- und Kleinschreibung.

b) Finden Sie mit *-in* movierte Nomina im Plural, wobei auch die Variante des Binnen-*I* berücksichtigt wird.

c) Finden Sie gezielt alle Wörter mit Binnenmajuskel (Binnengroßschreibung bzw. Großbuchstaben im Wortinneren).

1.1.18 | Lösung

a) ANNIS-Anfrage:
`tok=/(O|o)+(H|h)+/`
CQP-Anfrage:
`[word="(O|o)+(H|h)+"]`
TIGERSearch-Anfrage:
`[word=/(O|o)+(H|h)+/]`

b) ANNIS-Anfrage:
`lemma=/.+(I|i)n/`
Wortformsuche: `tok=/.+(I|i)(n|nnen)/`
CQP-Anfrage:
`[lemma=".+(I|i)n"]`
Wortformsuche: `[word=".+(I|i)(n|nnen)"]`
TIGERSearch-Anfrage:
`[lemma=/.+(I|i)n/]`
Wortformsuche: `[word=/.+(I|i)(n|nnen)/]`
Hinweis zu allen drei Suchanfragen: Sie finden sehr viele Treffer wie »ein« zusätzlich zu den gewünschten. Wie Sie gezielt bestimmte Lemmata, Wortarten usw. ausschließen können, erfahren Sie in den nachfolgenden Kapiteln.

c) ANNIS-Anfrage:
`tok=/[A-ZÄÖÜ][a-zöäüß]+[A-ZÄÖÜ][a-zöäüß]+/`
CQP-Anfrage:
`[word="[A-ZÄÖÜ][a-zöäüß]+[A-ZÄÖÜ][a-zöäüß]+"]`
TIGERSearch-Anfrage:
`[word=/[A-ZÄÖÜ][a-zöäüß]+[A-ZÄÖÜ][a-zöäüß]+/]`

Arbeitsaufgabe 3.1.2.11

Formulieren Sie Suchen mit den folgenden optionalen Elementen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Lemmata, die die Bestandteile *-er*, *-lich* und *-keit* in dieser Reihenfolge enthalten, wobei *-lich* nicht vorkommen muss.

b) Finden Sie Wörter (Wortformen), die auf *-steuer* oder *-steuern* enden und deren Erstbestandteil auf *-t*, *-n* oder *-g* endet. Berücksichtigen Sie, dass zwischen diesen Bestandteilen ein Fugen-*s* stehen kann, aber nicht muss. Die Treffermenge soll beide Varianten enthalten, sofern im Korpus vorhanden.

Lösung

a) ANNIS-Anfrage:

```
lemma=/.*er.*(lich)?.*keit.*/
```

CQP-Anfrage:

```
[lemma=".*er.*(lich)?keit.*"]
```

TIGERSearch-Anfrage:

```
[lemma=/.*er.*(lich)?keit.*/]
```

a) ANNIS-Anfrage:

```
tok=/.*(t|n|g)s?steuern?/
```

CQP-Anfrage:

```
[word=".*(t|n|g)s?steuern?"]
```

TIGERSearch-Anfrage:

```
[word=/.*(t|n|g)s?steuern?/]
```

Arbeitsaufgabe 3.1.2.12

Formulieren Sie Suchen mit Zeichen, die als reguläre Ausdrücke interpretiert werden könnten, für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie beliebig viele Abfolgen des Punkts (innerhalb desselben Tokens).

b) Finden Sie mit einem Punkt abgekürzte Wörter.

c) Finden Sie Asteriske (Sternchen) innerhalb der im Korpus verarbeiteten Texte. Diese können entweder alleine stehen oder Teile von Wörtern sein.

Lösung

a) ANNIS-Anfrage:

```
tok=/\.+/
```

CQP-Anfrage:

```
[word="\.+"]
```

TIGERSearch-Anfrage:

```
[word=/\.+/]
```

b) ANNIS-Anfrage:

```
tok=/[^.]+\./
```

ohne Zahlen: `tok=/[^.0-9]+\./`

CQP-Anfrage:

```
[word="[^.]+\."]
```

ohne Zahlen: `[word="[^.0-9]+\."]`
TIGERSearch-Anfrage:
`[word=/[^.]+\./]`
ohne Zahlen: `[word=/[^.0-9]+\./]`

c) ANNIS-Anfrage:
`tok=/.*\*.*/`
CQP-Anfrage:
`[word=".*\*.*"]`
TIGERSearch-Anfrage:
`[word=/.*\*.*/]`

Arbeitsaufgabe 3.1.2.14

Formulieren Sie Suchen nach mehreren Merkmalen bei demselben Token für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Vorkommen des Worts (Lemmas) *an*, die abgetrennte Verbpartikeln sind.
b) Finden Sie Vorkommen von *meinen*, die Possessivartikel (nach STTS: attributive Possessivpronomen) sind.
c) Finden Sie Vorkommen von *einen*, die Artikel sind.
d) Finden Sie Vorkommen von *meinen* und *einen*, die Verben sind.

Lösung

a) ANNIS-Anfrage:
`lemma="an" _=_ pos="PTKVZ"`
CQP-Anfrage:
`[lemma="an" & pos="PTKVZ"]`
TIGERSearch-Anfrage:
`[lemma="an" & pos="PTKVZ"]`

b) ANNIS-Anfrage:
`tok="meinen" _=_ pos="PPOSAT"`
CQP-Anfrage:
`[word="meinen" & pos="PPOSAT"]`
TIGERSearch-Anfrage:
`[word="meinen" & pos="PPOSAT"]`

c) ANNIS-Anfrage:
`tok="einen" _=_ pos="ART"`
CQP-Anfrage:
`[word="einen" & pos="ART"]`
TIGERSearch-Anfrage:
`[word="einen" & pos="ART"]`

d) ANNIS-Anfrage:
`tok=/m?einen/ _=_ pos=/V.*/`
CQP-Anfrage:
`[word="m?einen" & pos="V.*"]`
TIGERSearch-Anfrage:
`[word=/m?einen/ & pos=/V.*/]`

Arbeitsaufgabe 3.1.2.15

Formulieren Sie Suchen nach Abfolgen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Abfolgen von Pronomina und Nomina.
b) Finden Sie die Negationspartikel *nicht* am Satzende (also vor einem Satzbeendungszeichen).
c) Finden Sie die Konjunktion *und* direkt nach Kommata.
d) Finden Sie Abfolgen des Lemmas *auf*, des Lemmas *jeder* oder *kein* und eines beliebigen Nomens.

1.1.22 | Lösung

a) ANNIS-Anfrage:

```
pos=/P.*(AT|S)/ . pos=/N./
```

CQP-Anfrage:

```
[pos="P.*(AT|S)"] [pos="N."]
```

TIGERSearch-Anfrage:

```
[pos=/P.*(AT|S)/] . [pos=/N./]
```

b) ANNIS-Anfrage:

```
pos="PTKNEG" . pos=/\$\./
```

CQP-Anfrage:

```
[pos="PTKNEG"] [pos="\$\."]
```

TIGERSearch-Anfrage:

```
[pos="PTKNEG"] . [pos=/\$\./]
```

c) ANNIS-Anfrage:

```
pos=/\$,/ . lemma="und"
```

CQP-Anfrage:

```
[pos="\$,"] [lemma="und"]
```

TIGERSearch-Anfrage:

```
[pos=/\$,/] . [lemma="und"]
```

d) ANNIS-Anfrage:

```
lemma=/(auf|jeder|kein)/ . pos=/N./
```

CQP-Anfrage:

```
[lemma="(auf|jeder|kein)"] [pos="N."]
```

TIGERSearch-Anfrage:

```
[lemma=/(auf|jeder|kein)/] . [pos=/N./]
```

Arbeitsaufgabe 3.1.2.16

Formulieren Sie Suchen nach ausgeschlossenen Elementen für die einzelnen Suchsysteme. Testen Sie die Ergebnisse an den verschiedenen Korpora, die Sie über die Suchsysteme abfragen können.

a) Finden Sie Lemmata, die auf *-en* enden, aber keine Verben sind.
b) Finden Sie Wortformen, die kein *e* oder *E* enthalten.
c) Finden Sie Abfolgen von Artikel (nach STTS-Definition), einem Element und Nomen, wobei das mittlere Element kein Adjektiv sein soll.

Lösung

a) ANNIS-Anfrage:

```
lemma=/.*en/ _=_ pos!=/V.*/
```

CQP-Anfrage:

```
[lemma=".*en" & pos!="V.*"]
```

TIGERSearch-Anfrage:

```
[lemma=/.*en/ & pos!=/V.*/]
```

b) ANNIS-Anfrage:

`tok=/[^Ee]*/` oder `tok!=/.*[Ee].*/`

CQP-Anfrage:

`[word="[^Ee]*"]` oder `[word!=".*[Ee].*"]`

TIGERSearch-Anfrage:

`[word=/[^Ee]*/]` oder `[word!=/.*[Ee].*/]`

c) ANNIS-Anfrage:

```
pos="ART" . pos!=/ADJ./ . pos=/N./
```

CQP-Anfrage:

```
[pos="ART"] [pos!="ADJ."] [pos="N."]
```

TIGERSearch-Anfrage:

```
[pos="ART"] . #1:[pos!=/ADJ./] &
#1 . [pos=/N./]
```

Arbeitsaufgaben 3.1.2.18

1. Formulieren Sie für die CQP-Anfragesprache eine Suche nach einer Präpositionalphrase mit postnominalem (nachgestelltem) *wegen* im Vorfeld eines Satzes, indem Sie in der Anfrage die folgende Abfolge festlegen: Satzbeendungszeichen – optionales Artikelwort (»ART« oder »P.*AT« nach STTS) – optionales Adjektiv – Nomen (»NN« oder »NE« nach STTS) – Lemma *wegen* – finites Verb.
2. Formulieren Sie analog dazu eine Suchanfrage mit präpositionalem (vorangestelltem) *wegen*.
   Führen Sie die Suchen für das Korpus »DeWaC 1« im CQP-Interface unter korpling.german.hu-berlin.de/cqpwi/ durch und vergleichen Sie die Treffer für die unterschiedlichen Anfragen. Vergleichen Sie die Aufgabe 3 in Kap. 4.6.1 für eine Auswertung der Suchergebnisse.

Lösungen

1.

```
[pos="\$\."] [pos="(ART|P.*AT)"]? [pos="ADJA"]?
[pos="N(N|E)"] [lemma="wegen"] [pos="V.FIN"]
```

→ 72 Treffer in »DeWaC 1«

Hinweis: Anstatt des Fragezeichen-Operators bei dem Suchelement für pränominale Adjektive können Sie – das ist dann noch präziser – auch ein Sternchen für beliebig viele Vorkommen inklusive keinem Vorkommen setzen.

2. `[pos="\$\."] [lemma="wegen"] [pos="(ART|P.*AT)"]? [pos="ADJA"]? [pos="N(N|E)"] [pos="V.FIN"]`
→ 2049 Treffer in »DeWaC 1«
Hinweis: Anstatt des Fragezeichen-Operators bei dem Suchelement für pränominale Adjektive können Sie – das ist dann noch präziser – auch ein Sternchen für beliebig viele Vorkommen inklusive keinem Vorkommen setzen.

Arbeitsaufgaben 3.1.2.19

1. Die folgenden Suchanfragen beziehen sich zunächst auf die Spannen-Annotationen in der Abb. 3.8. Nutzen Sie die dort abgebildeten Variablen und Werte. Sie können diese Suchanfragen nicht überprüfen, weil das Korpusbeispiel fingiert ist.
   a) Finden Sie alle Fälle, in denen (wie in der Abbildung) nach der linken Satzklammer unmittelbar ein Dativobjekt folgt.
   b) Finden Sie alle Fälle, in denen (wie in der Abbildung) das Mittelfeld ein Adverbial enthält.
   c) Finden Sie entgegen der Abbildung alle Fälle, in denen das Mittelfeld nur genau ein Adverbial enthält, also mit ihm deckungsgleich ist.
   Welche der oben vorgestellten Operatoren, die Sie zwischen die Suchelemente setzen können, finden diese Fälle auch (und zusätzlich weitere)?

2. Die folgenden Suchaufgaben beziehen sich auf das BeMaTaC-Korpus BeMaTaC_L1_3.0, in der ANNIS-Instanz https://korpling.german.hu-berlin.de/annis3/intro. Dieses Korpus besteht aus transkribierten gesprochenen Dialogen, die mit diversen Annotationen angereichert wurden. Entnehmen Sie die für die Suche notwendigen Variablen- und Wertenamen den Aufgaben, formulieren Sie die jeweilige Suche und überprüfen Sie die Suche anhand des BeMaTaC-Korpus.
   In dem Korpus sind Äußerungen auf einer Annotationsebene »utt« (für »utterance«) und als Spannen mit der Bezeichnung »utt« annotiert. Die transkribierten und normalisierten Wortformen sind wie in den bisher behandelten Korpora mit Lemma- (Ebene: »lemma«) und Wortartenannotationen (Ebene: »pos«; STTS-Tagset) versehen.
   a) Finden Sie äußerungsinitiale Verben (Verben, die am Anfang einer »utt«-Spanne stehen).
   Hinweis: Stellen Sie vor dem Abschicken der Suche den linken Trefferkontext (unter dem Reiter »Search Options« in ANNIS) auf null, um eine gute Trefferansicht zu erhalten.
   b) Finden Sie Formen des Lemmas *gehen* am Ende von Äußerungen.
   Hinweis: Stellen Sie vor dem Abschicken der Suche den linken Trefferkontext auf fünf und den rechten auf null, um eine gute Trefferansicht zu erhalten.

Lösungen

1. a) `TopFeld="LSK" . Satzfunktion="OBJD"`
b) `TopFeld="MF" _i_ Satzfunktion="ADV"`
c) `TopFeld="MF" _=_ Satzfunktion="ADV"`
Diese Fälle werden auch durch die Anfragen
`TopFeld="MF" _l_ Satzfunktion="ADV"`
und
`TopFeld="MF" _r_ Satzfunktion="ADV"`
gefunden.

2. a) `utt _l_ pos=/V.*/`
b) `utt _r_ lemma="gehen"`

Arbeitsaufgabe 3.1.2.20

1. Verwenden Sie die oben genannten Informationen zum TIGER-Korpus und formulieren Sie Suchen für das TIGER-Korpus, wenn es im ANNIS-Suchinterface durchsucht wird sowie wenn es in TIGERSearch durchsucht. Sie können für Anfragen in der TIGERSearch-Anfragesprache auch die Online-Suchinstanz unter der Webadresse http://fnps.coli.uni-saarland.de:8080/ verwenden.
a) Finden Sie alle Präpositionalphrasen.
b) Finden Sie alle koordinierten Sätze.

Lösung

a) `[cat="PP"]`
b) `[cat="CS"]`

Arbeitsaufgaben 3.1.2.21

1. Die folgenden Suchaufgaben beziehen sich auf das TIGER-Korpus. Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2FpJDIU) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. TüNDRA.
a) Finden Sie alle Fälle, in denen eine NP unmittelbar das Lemma *Herz* dominiert.
b) Finden Sie alle Fälle, in denen eine NP unmittelbar eine AP dominiert. (APn werden im TIGER-Korpus nur annotiert, wenn das Kopfadjektiv erweitert ist.)
c) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar ein pränominales Adjektiv und eine weitere NP enthält. Beachten Sie, dass die Suche in ANNIS hypothetisch ist, da das Korpus nur in TüNDRA und als lokal in TIGERsearch zu installierende Datei verfügbar ist.
Hinweis: Formulieren Sie diese Anfrage nur für das ANNIS-Suchinterface, wenn Sie keinen Zugriff auf eine lokale Installation von TIGER Search haben.

2. Die folgenden Suchaufgaben beziehen sich auf das TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL). Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2U1EZck) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. TüNDRA. (Bitte beachten Sie, dass die Suchanfrage in ANNIS hypothetisch ist, da das Korpus dort nicht verfügbar ist.)
   a) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar eine Adjektivphrase dominiert. (Adjektivphrasen können im TüBa-D/Z-Korpus auch aus einer Worteinheit bestehen.)
   b) Finden Sie alle Fälle, in denen eine Adjektivphrase über genau zwei Generationen das Lemma *sehr* dominiert.

Lösungen

1. a) Suchanfrage für ANNIS (Korpus »tiger2«):

```
cat="NP" > lemma="Herz"
```

Suchanfrage für TüNDRA/TIGERSearch:

```
[cat="NP"] > [lemma="Herz"]
```

Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus.

b) Suchanfrage für ANNIS (Korpus »tiger2«):

```
cat="NP" > cat="AP"
```

Suchanfrage für TüNDRA/TIGERSearch:

```
[cat="NP"] > [cat="AP"]
```

Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus.

c) Suchanfrage für ANNIS (Korpus »tiger2«):

```
cat="NP" > pos="ADJA" &
#1 > cat="NP"
```

Suchanfrage für TüNDRA/TIGERSearch:

```
#1:[cat="NP"] > [pos="ADJA"]&
#1 > [cat="NP"]
```

Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus. Das Interface unter der Adresse http://fnps.coli.uni-saarland.de:8080/ kann die Variablen nicht verarbeiten.

2. a) Suchanfrage für TüNDRA/TIGERSearch (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA):

```
[cat="NX"] > [cat="ADJX"]
```

b) Suchanfrage für TüNDRA/TIGERSearch (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA):

```
[cat="ADJX"] >2 [lemma="sehr"]
```

Arbeitsaufgaben 3.1.2.22

1. Die folgenden Suchaufgaben beziehen sich auf das TIGER-Korpus. Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2FpJDIU) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch.
   a) Finden Sie alle Fälle, in denen eine VP unmittelbar eine NP als Akkusativobjekt dominiert.
   b) Finden Sie alle Fälle, in denen eine NP unmittelbar eine AP als Vergleichsphrase (Kantenbezeichnung: »CC«) dominiert.
   c) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar ein pränominales Adjektiv und eine weitere NP als Genitivattribut (Kantenbezeichnung: »AG«) enthält.
2. Das im ANNIS-Suchinterface durchsuchbare Korpus »pcc2.1« (das Potsdamer Zeitungskommentarkorpus; https://bit.ly/2Y9Rz8C) enthält ebenso Phrasenstrukturannotationen, welche nach dem TIGER-Korpus erstellt wurden.
   Finden Sie in diesem Korpus alle Fälle, in denen eine VP eine PP als Präpositionalobjekt unmittelbar dominiert.
3. Die folgenden Suchaufgaben beziehen sich auf das TüBa-D/Z-Korpus (https://bit.ly/2ulbCmL). Nehmen Sie die Liste mit Phrasen- und Kantenbezeichnungen des Korpus zur Hand (https://bit.ly/2U1EZck) und formulieren Sie Suchen für das Korpus in ANNIS und TIGERSearch bzw. TüNDRA.
   a) Finden Sie alle Fälle, in denen eine Nominalphrase unmittelbar eine Adjektivphrase dominiert. (Adjektivphrasen können im TüBa-D/Z-Korpus auch aus einer Worteinheit bestehen.)
   b) Finden Sie alle Fälle, in denen ein Akkusativobjekt im Vorfeld steht. (Die Suchanfrage muss so ausgedrückt werden, dass das Vorfeld eine beliebige Konstituente mit der Funktion Akkusativobjekt dominiert.)

Lösungen

1. a) Suchanfrage für ANNIS (Korpus »tiger2«):
   ```
   cat="VP" >[func="OA"] cat="NP"
   ```
   Suchanfrage für TüNDRA/TIGERSearch (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA):
   ```
   [cat="VP"] >OA [cat="NP"]
   ```
   Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus.

   b) Suchanfrage für ANNIS (Korpus »tiger2«):
   ```
   cat="NP" >[func="CC"] cat="AP"
   ```
   Suchanfrage für TüNDRA/TIGERSearch:
   ```
   [cat="NP"] >CC [cat="AP"]
   ```
   Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus.

   c) Suchanfrage für ANNIS (Korpus »tiger2«):
   ```
   cat="NP" > pos="ADJA" &
   #1 >[func="AG"] cat="NP"
   ```
   Suchanfrage für TüNDRA/TIGERSearch:
   ```
   #NP:[cat="NP"] > [pos="ADJA"] &
   #NP >AG [cat="NP"]
   ```

Hinweis: Diese Anfrage gilt für das TIGER-Korpus in TIGERSearch. In TüNDRA existiert (derzeit) keine Instanz des Korpus. Das Interface unter der Adresse http://fnps.coli.uni-saarland.de:8080/ kann die Variablen nicht verarbeiten.

2. Suchanfrage für ANNIS (Korpus »tiger2«):

```
cat="VP" >[func="OP"] cat="PP"
```

Hinweis: Diese Suchanfrage funktioniert gleichermaßen für das in ANNIS vorhandene Korpus »tiger2«.

3. a) Suchanfrage für ANNIS:

```
cat="NX" > cat="ADJX"
```

Hinweis: Das TüBa-D/Z-Korpus ist in ANNIS nicht verfügbar.
Suchanfrage für TüNDRA/TIGERSearch (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA):

```
[cat="NX"] > [cat="ADJX"]
```

b) Suchanfrage für ANNIS:

```
cat="VF" >OA cat
```

Hinweis: Das TüBa-D/Z-Korpus ist in ANNIS nicht verfügbar.
Suchanfrage für TüNDRA/TIGERSearch (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA):
`[cat="VF"] >OA [cat=/.*/]` (für die Korpora »TüBa-D/Z 10« und »TüBa-D/Z v11« in TüNDRA)

Arbeitsaufgabe 3.1.2.24

Die folgenden Suchaufgaben beziehen sich auf das Korpus »tiger_dep_v2.2« im ANNIS-Suchinterface sowie das Korpus »TüBa-D/Z v10 Dependency (Experimental)« im TüNDRA-Suchinterface.

a) Finden Sie alle Vorkommen des Lemmas *schön*, die unmittelbar von einem Nomen dominiert werden.

b) Finden Sie Adverbien, die unmittelbar von Verben dominiert werden.

Lösung

a) Suchanfrage für ANNIS:

```
pos="NN" ->dep lemma="schön"
```

Suchanfrage für TüNDRA/TIGERSearch:

```
[pos="NN"] > [lemma="schön"]
```

a) Suchanfrage für ANNIS:

```
pos=/V.*/ ->dep pos="ADV"
```

(Einschränkung auf Vollverben: `pos=/VV.*/ ->dep pos="ADV"`)
Suchanfrage für TüNDRA/TIGERSearch:

```
[pos=/V.*/] > [pos="ADV"]
```

(Einschränkung auf Vollverben: `[pos=/VV.*/] > [pos="ADV"]`)

Arbeitsaufgabe 3.1.2.25

Die folgenden Suchaufgaben beziehen sich auf das Korpus »tiger_dep_v2.2« im ANNIS-Suchinterface sowie das Korpus »TüBa-D/Z v10 Dependency (Experimental)« im TüNDRA-Suchinterface.

a) Finden Sie alle Nomina, die andere Elemente als Akkusativobjekte unmittelbar dominieren. (Das Funktionslabel für Akkusativobjekte im TIGER-Korpus ist »OA«, im TüBa-D/Z-Dep-10-Korpus »OBJA«.)

b) Finden Sie alle Präpositionen, die unmittelbar von Nomina als (postnominale) Modifikatoren abhängen. (Das Funktionslabel für postnominale Modifikatoren im TIGER-Korpus ist »MNR«, im TüBa-D/Z-Dep-10-Korpus »PP«.)

Lösung

a) Suchanfrage für ANNIS:

```
pos="NN" ->dep[deprel="OA"] node
```

Suchanfrage für TüNDRA/TIGERSearch:

```
[pos="NN"] >OBJA [pos=/.*/]
```

Hinweis: Die Suchanfragen liefern fast ausschließlich elliptische Fälle, in denen das Verb fehlt und deshalb Subjekt und Objekt direkt miteinander verknüpft wurden.

b) Suchanfrage für ANNIS:

```
pos="NN" ->dep[deprel="MNR"] pos="APPR"
```

Suchanfrage für TüNDRA/TIGERSearch:

```
[pos="NN"] >PP [pos="APPR"]
```

Arbeitsaufgabe 3.1.2.26

Testen Sie die angegebenen Suchanfragen mit den genannten wortartmäßigen Varianten zu verschiedenen Koreferenzausdrücken im PCC-Korpus, das Sie über das ANNIS-Suchinterface abfragen können. Überlegen Sie, welche Wortarten zusätzlich zu den genannten koreferent sein können, und testen Sie die Annahmen am genannten Korpus.

Lösung

Um zu testen, welche einzelnen Wortarten koreferent mit vorangegangenen Ausdrücken sind, können Sie die Suchanfrage

```
node ->anaphor_antecedent node _=_ pos
```

verwenden und nach der dritten Suchvariable »pos« quantitativ auswerten (Funktion »More« > »Frequency Analysis«, dann die ersten beiden Variablen »node« und »node« löschen und alle koreferenten Wortarten auflisten lassen). Mit Blick auf das Ergebnis liegen viele Kategorien auf der Hand, allerdings ist die Kategorie »ADJA« zunächst eine gewisse Überraschung. Testen Sie diese Kategorie mit der Anfrage

```
node ->anaphor_antecedent node _=_ pos=" ADJA"
```

und Sie sehen, welche Fälle sich dahinter verbergen.

Arbeitsaufgaben 3.1.2.27

1. In der Korpussuchmaschine ANNIS finden Sie das Korpus »Maerchenkorpus«, bestehend aus 211 Märchen und Kinderlegenden der Gebrüder Grimm. Sie sind mit dem üblichen automatischen Verfahren getaggt, so dass sie nach Lemmata und STTS-Wortarten durchsuchbar sind. Die einzelnen Märchen sind in der Suche mit der Metadaten-Variable »Titel« voneinander trennbar. Zwei der Märchentitel sind »Sneewittchen« und »Daumesdick«.
   a) Finden Sie alle Fälle des Lemmas *Apfel* in dem Märchen »Sneewittchen«.
   b) Führen Sie dieselbe Suche ohne Metadateneinschränkung durch und schauen Sie, ob das Nomen überhaupt außerhalb des Schneewittchen-Märchens in der Märchensammlung auftritt.
   c) Finden Sie alle Wörter in dem Märchen »Daumesdick«, die gemäß dem STTS-Tagset als Adverbien ausgewiesen sind.
2. (zu Szenario 1): Wenden Sie die eingeführte Suche auf die einzelnen Jahre von 1996 bis 2002 an und schauen Sie nach offensichtlichen Veränderungen in der Verwendung über die Jahre hinweg.
3. (zu Szenario 2): Vergleichen Sie die Vorkommen Nomina in den Fachgebieten Chemie (»chemie«), Physik (»physik«) und Medizin (»medizin«) und schauen Sie, ob Sie qualitative Unterschiede ausmachen können.
4. (zu Szenario 3):
   a) Vergleichen Sie die Vorkommen von Modalpartikeln in den genannten Falko-Lernergruppen L1 Englisch, Dänisch, Russisch und Französisch im Korpus »falkoEssayL2v2.4« des ANNIS-Suchinterfaces.
   b) Schauen Sie sich anschließend die Treffer im Korpus »falkoEssayL1v2.3« an. Sie müssen hier den Metadatenzusatz weglassen, weil das Vergleichskorpus ausschließlich aus deutschsprachigen Muttersprachlerinnen und Muttersprachlern besteht und deshalb für die Variable »L1« bzw. »Muttersprache« keine Metadatenannotation benötigt wird.

   Hinweis: Auch an dieser Stelle dürfen Sie nur qualitativ und nicht quantitativ vergleichen, weil Sie Datenmenge der jeweiligen Lernerkohorte nicht kennen und deshalb nicht einschätzen können, ob die Anzahl der gefundenen Treffer gemessen an der jeweiligen Gesamtdatenmenge hoch oder niedrig ist. Vergleichen Sie zum Konzept des Normalisierens Kap. 4.5.1.

Lösungen

1. a) `lemma="Apfel" & meta::Titel="Sneewittchen"`
   b) Ja: Während in »Sneewittchen« nur 6 Treffer vorliegen, zählt das gesamte Korpus 63 Treffer (in 13 Märchen).
   (Suche: `lemma="Apfel"`)
   c) `pos="ADV" & meta::Titel="Daumesdick"` (161 Treffer)
2. 1996: 1 Treffer
   1997: 8 Treffer
   1998: 2 Treffer
   1999: 17 Treffer
   2000: 27 Treffer
   2001: 32 Treffer
   2002: 23 Treffer

Während in den früheren Jahrgängen der Suchbegriff *E-Mail* vorwiegend mit den Themen »Sicherheit«, »Bildung« und »telekommunikative Modernisierung« verknüpft ist (*Da gibt es zum einen die Freaks , denen Begriffe wie Homepage , **E-Mail** , Browser -- und was es da so alles gibt -- ganz locker über die Lippen gehen und die überhaupt nicht verstehen können , daß jemand damit nichts anfangen kann .* (...); Jahrgang 1997, Treffer 3), wird der Begriff zunehmend im Sinne eines selbstverständlichen Kommunikationsmediums gebraucht (*Viele **E-Mails** , die ich im Zusammenhang mit der furchtbaren Flutkatastrophe bekommen habe , zeugen davon ,* (...); Jahrgang 2002, Treffer 19).

3. Suchanfrage z. B.: `[pos="NN"] ::match.abstract_sachgebiet="medizin"`; stellen Sie die Ausgabe auf »frequencies« und sortieren Sie nach Lemmata. In »medizin« springen z. B. die Lemmata »Patient«, »Therapie«, »Behandlung« ins Auge, die sich in den anderen beiden Fachgebieten nicht unter den frequentesten Wörtern befinden. »Arbeit« und »Untersuchung« sind für die drei Fachbereiche unspezifische bzw. verbindende Begriffe, mit denen auf das angestrebte Projekt referiert wird. Häufige Begriffe in »chemie« wie »Enzym« und »Protein« legen eine fachliche Nähe zum biologischen Fachbereich nahe, der sich mit Blick auf das Metadatum »biologie« belegen lässt (auch hier sind diese Begriffe besonders häufig). Im Bereich »physik« springen dominierende Begriffsfelder zur experimentellen Methodik wie »Experiment«, »Methode«, »Modell«, »Detektor« ins Auge, die deutlich vor inhaltlichen Begriffen (»Temperatur«, »Elektron«) gerankt sind.
4. a) und b) In jeder der Lernergruppen finden sich einzelne Belege für die verschiedenen Modalpartikeln, vor allem für »ja« und »wohl«, was den muttersprachlichen Vergleichsdaten entspricht. Die dänische Lernerkohorte nutzt die Modalpartikeln praktisch wie die muttersprachliche Vergleichsgruppe, während die übrigen drei Lernergruppen die Lexeme deutlich seltener verwenden (hierfür muss man die absoluten Vorkommen an den jeweiligen Gesamtgrößen der Gruppen messen, siehe Kap. 4.5.1 zur Normalisierung). Zum Beispiel kommen innerhalb der gesamten L1-Russisch-Kohorte nur zwei zielsprachige Fälle (einer von »wohl«, einer von »ja«) vor. (Ein weiterer Fall von »wohl« betrifft das Adjektiv, ein weiterer Fall von »wohl« ist ungrammatisch und betrifft entweder das Adjektiv oder die Modalpartikel). Somit sticht die dänische Lernergruppe im Modalpartikelgebrauch als sehr zielsprachlich hervor.

Arbeitsaufgaben 3.2.1

1. Formulieren Sie eine Suche im DWDS-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.
2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.
3. Testen Sie diese Suchen an dem »DWDS-Kernkorpus (1900–1999)« in der Funktion der DWDS-Korpusbelege (http://www.dwds.de/r).

Lösungen

1. `"dank with $p=APPR @des"`
2. `"{dank,gemäß,trotz} with $p=APPR @des"`
3. Die Suche zu Aufgabe 1 ergibt 127 Treffer. Die Suche zu Aufgabe 2 ergibt 1722 Treffer.

Arbeitsaufgaben 3.2.2

1. Formulieren Sie eine Suche im COSMAS-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.
2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.
3. Testen Sie diese Suchen an dem TAGGED-T-Archiv des COSMAS-II-Interfaces.

Lösung

1. `trotz ODER dank ODER gemäß /w0 MORPH(AP pr) /+w1 des`
2. `trotz dank gemäß /w0 MORPH(AP pr) /+w1 des` , »logisches ›ODER‹« aktivieren
3. Die Suche zu Aufgabe 1 ergibt 26800 Treffer. Die Suche zu Aufgabe 2 ergibt 38796 Treffer.

Arbeitsaufgaben 3.2.3

1. Formulieren Sie eine Suche im DGD-Suchsystem, die Ihnen Belege für die unmittelbare Abfolge des Lemmas *dank* als Präposition und der Wortform *des* liefert.
2. Formulieren Sie anschließend eine Suchanfrage, die zusätzlich zu *dank* auch die Präpositionen *gemäß* und *trotz* findet.
3. Testen Sie diese Suchen an dem FOLK-Korpus im Suchinterface.

Lösungen

1. Eingabefeld »»Lemma«: `dank` , Eingabefeld »POS«: `APPR`
   Dann Kontext: 1 Token, rechts: Eingabefeld »Normalisiert«: `des`
1. Eingabefeld »»Lemma«: `(dank|gemäß|trotz)` , »Reguläre Ausdrücke« aktivieren
2. Suche zu Aufgabe 1 ergibt 6 Treffer ohne Kontextfilterung, 1 Treffer mit Kontextfilterung.
   Suche zu Aufgabe 2 ergibt 47 Treffer ohne Kontextfilterung, 5 Treffer mit Kontextfilterung.
   (Die geringen Zahlen hängen wahrscheinlich mit Aufbereitungsfehlern zusammen.)

Arbeitsaufgaben 3.3.1

1. Sie möchten Typen von *-bar*-Adjektiven, die aus Verbstämmen und dem Ableitungssuffix *–bar* gebildet wurden, in einem Korpus finden, indem Sie nach Lemmata suchen, deren Form auf *–bar* endet (CQP-Suchausdruck z. B. [lemma = ".*bar"]).
   - Ihre Ergebnisliste ist:
     anwendbar, begehbar, Eisbar, verhandelbar, Barbar, brennbar, kämmbar, denkbar, Nachbar, furchtbar, zerstörbar, bar, offenbar, greifbar
   - Ermitteln Sie für dieses Suchergebnis den Wert für Precision und geben Sie somit an, wie genau die Suche hinsichtlich der gefundenen Treffer ist.
   - *Zusatzaufgabe:* Überlegen Sie auf der Grundlage der genannten Treffermenge, wie Sie die Suchanfrage präzisieren können, damit sich ein höherer Wert für Precision ergibt.
2. Sie haben bei der Suche nach einem beliebigen linguistischen Phänomen rund 12000 Treffer erzielt, die Sie nicht alle manuell überprüfen können. Sie ziehen deshalb aus der Gesamtmenge an Treffern eine Stichprobe von 200 Treffern und evaluieren diese auf Korrektheit. Dabei ermitteln Sie 37 Fehler (Fälle, die eigentlich nicht in die Treffermenge gehören).
   Ermitteln Sie den Wert für Precision.

Lösungen

1. Falsche Treffer sind: *Eisbar*, *Barbar*, *Nachbar*, *furchtbar* (*Furcht* ist kein Verb), *bar*, *offenbar*
   Precision: 6/14 ≈0,43
   - *Zusatzaufgabe:* Man kann *bar* durch den »+«-Operator ausschließen und großgeschriebene Lemmata ausschließen. Die Suchanfrage lautet dann in CQP:
     [lemma = ". + bar" & lemma! = "[A-ZÖÄÜ].*"]
     oder auch
     [lemma = "[^A-ZÖÄÜ].*bar"]
2. Precision: 163/200 = 0,815

Arbeitsaufgabe 3.3.2

Sie möchten mithilfe einer Wortartensuche finite Verbformen aus einem Korpus extrahieren (CQP-Suchausdruck z. B. [pos = "V.*FIN"]). In einem Korpusausschnitt mit 200 als finite Verben getaggten Wortformen stellen Sie fest, dass sechs Treffer falsche Treffer sind, außerdem wurden acht finite Verben nicht als solche (sondern als infinit oder andere Wortarten) erkannt.
Ermitteln Sie für dieses Suchergebnis den Wert für Recall und geben Sie somit hinsichtlich des überprüften Korpusausschnitts an, wie genau die Suche hinsichtlich der eigentlich zu findenden Elemente im Korpus ist.

Lösung

Recall: 194/202 ≈0,96

Arbeitsaufgabe 3.3.3

Sie möchten in einem wortartgetaggten Korpus die Fokuspartikeln *auch*, *schon* und *nur* untersuchen und formulieren dazu eine Suche nach den drei Lemmata, gefolgt von Artikelwörtern, Präpositionen, pränominalen Adjektiven, Nomina oder Eigennamen (CQP-Suchausdruck z. B. [lemma = "(auch|schon|nur)"][pos = "(ART|P.*AT|APPR|ADJA|N(E|N)"]). Innerhalb von 100 Treffern ermitteln Sie, dass es sich bei 14 Fällen um keine Fokuspartikeln handelt, weil das gemeinsame Auftreten der beiden gesuchten Token unabhängig voneinander erfolgt und die Lexeme *auch*, *schon* und *nur* somit als Adverbiale zu interpretieren sind. Sie stellen außerdem fest, dass innerhalb der untersuchten Textmenge acht Fälle von *auch*, *schon* und *nur* auftreten, in denen es sich um Fokuspartikeln handelt, die allerdings nicht linksadjazent zur Bezugskonstituente stehen oder vor Pronomina auftreten.

- Berechnen Sie für die besagte Suchanfrage innerhalb des analysierten Korpusausschnitts den F-score und geben Sie somit für den untersuchten Korpusausschnitt einen Wert für die allgemeine Suchgenauigkeit an.

Lösung

$$F = \frac{2(86/100 \times 86/94)}{86/100 + 86/94} \approx 0{,}89$$

Arbeitsaufgaben 4.1

1. Bearbeiten Sie die unter der Webadresse https://bit.ly/2TO2juR verfügbaren Exportdaten so, dass nur die beiden relevanten Wörter innerhalb der Spitzklammern übrig bleiben. Gehen Sie dabei folgendermaßen vor:
   - Laden Sie die angegebene Datei auf Ihren Computer.
   - Öffnen Sie die Datei in einem Texteditor, der beim Suchen und Ersetzen reguläre Ausdrücke unterstützt, z. B. Notepad++ (https://notepad-plus-plus.org/).
   - Öffnen Sie die Funktion »Replace« (bzw. »Ersetzen«; in Notepad++ zu erreichen mit STRG-f oder im Menü unter »Search« > »Replace«).
   - Geben Sie in der Suche an, dass Sie reguläre Ausdrücke verwenden (»Regular expression«).
   - Geben Sie in das Suchfenster ».*<« (ohne Anführungszeichen) ein. Geben Sie nichts in das Ersetzen-Fenster ein. Sie ersetzen somit sämtliche Zeichen einer Zeile vor der öffnenden Spitzklammer mit nichts, d. h. Sie löschen den String.
   - Setzen Sie den Cursor an die erste Position im Dokument und klicken Sie »Replace All«. (Wenn der Cursor irgendwo im Dokument steht, können Sie auch »Replace in All Opened Documents« betätigen.)
   - Der linke Trefferkontext bis zur öffnenden Spitzklammer wurde entfernt. Entfernen Sie nun den rechten Kontext, indem Sie das Ersetzen-Prozedere mit der Eingabe »>.*« wiederholen.
   - Nun sind lediglich ein Satzzeichen und ein Leerzeichen vor dem relevanten Suchtreffer vorhanden. Diese kann man mit dem Suchbefehl »\n. » löschen (das »\n« bezeichnet einen Zeilenumbruch bzw. -anfang,

der Punkt ein beliebiges Zeichen und das Leerzeichen wird wörtlich aufgefasst). Was übrig bleibt, ist die für die Auswertung relevante Zeichenkette.

2. Bearbeiten Sie die unter der Webadresse https://bit.ly/2HMjx4I verfügbaren Exportdaten so, dass eine sechsspaltige Tabelle mit der Kopfzeile wie oben gezeigt entsteht und die einzelnen Werte der Exportdatei korrekt zugeordnet sind. Gehen Sie dabei folgendermaßen vor:
- Gehen Sie bis zur Eingabe des Suchstrings vor wie in Aufgabe 1.
- Geben Sie im Suchfeld den Ausdruck » ',' » (einfaches Anführungszeichen, Komma, einfaches Anführungszeichen) ein. Geben Sie im Feld für die Ersetzung »\t« ein (die Kombination steht für einen Tabulator-Abstand).
- Kopieren Sie den Inhalt in ein geöffnetes Tabellenblatt einer LibreOffice-Calc-Datei oder in eine Microsoft-Excel-Datei (Zelle A1).
- Löschen Sie die Spalten A, E, G und H (die für die Vorgabe irrelevanten Spalten).
- Fügen Sie ganz oben im Dokument eine Zeile hinzu.
- Überschreiben Sie die Spalten mit den oben angegebenen Werten.
- Speichern Sie die Datei als Trennzeichen-getrennte Datei unter dem Namen »Konvertieren_2_konvertiert.csv«.

3. Bearbeiten Sie die in Aufgabe 2 erstellte Datei weiter, indem Sie die Werte der Spalten »Verb: Lemma« und »Obj: Lemma« mit einem Unterstrich verbunden zusammenführen und die Kopfzeile löschen. Gehen Sie hierzu wie folgt vor:
- Öffnen Sie die Datei, die Sie bei der Bearbeitung von Aufgabe 2 gespeichert haben. Die Datei können Sie auch unter der Webadresse https://bit.ly/2FpKqJS beziehen.
- Schreiben Sie in die Zelle G2 den Befehl »=VERKETTEN(B2;"_";E2)« (ohne Anführungszeichen) und betätigen Sie Enter.
- Markieren Sie die Zelle G2 und klicken Sie doppelt auf das Plus rechts unten in der markierten Zelle. Der Befehl wird bis in die letzte gefüllte Zeile kopiert.
- Kopieren Sie den Text von Zelle G2 bis G539 in Zelle H1 bis H540, ohne dass die Formeln mitkopiert werden (»Inhalte einfügen...« > »Werte« bzw. »Formeln« nicht anklicken).
- Löschen Sie alle Spalten bis auf Spalte H und speichern Sie das Ergebnis unter dem Namen »Konvertieren_3_konvertiert.csv«.

  Hinweis 1: Sie können die Konversion auch mit einem Texteditor wie Notepad++ durchführen, indem Sie die zwei relevanten Spalten dorthin kopieren und den Tabulatorabstand (»\t«) durch einen Unterstrich ersetzen.

  Hinweis 2: In Kap. 4.3 wird behandelt, wie man die Daten wie die in dieser Aufgabe erstellten nach Frequenzen auswertet. So kann man z. B. analysieren, dass die Verbindung »bringen_Frühstück« mit genau acht Vorkommen die häufigste Verb-Objekt-Verbindung in den zugrunde liegenden Korpusdaten ist (es handelt sich um ein kleines, sehr spezifisches Korpus mit Kafka-Texten).

Lösungen

1. https://bit.ly/2U1ckF3
2. https://bit.ly/2HY9B8k
3. https://bit.ly/2FyvIzf

Arbeitsaufgaben 4.5.1

1. Bestimmen Sie anhand der vorangegangenen Informationen und Ihrem linguistischen Wissen die Normalisierungsgröße in den folgenden Vergleichsszenarien: Es werden jeweils zwei (unterschiedlich große) Korpora auf die genannte linguistische Kategorie hin verglichen. An welcher Normalisierungsgröße müssen Sie die absoluten Werte bzw. ausgezählten Mengen messen, um vergleichbare Zahlen zu erhalten?
   a) Nebensätze
   b) Verb-erst-Sätze
   c) Relativsätze
   d) Nominalkomposita
   e) Schwa-Tilgungen
   f) Wortabbrüche
   g) kausale Diskursrelationen
   h) Perfektsätze
   i) In dialogischen Texten: Unterbrechungen
2. Rechnen Sie die absoluten Werte in vergleichbare Werte um: Gegeben sind jeweils absolute Werte zu einer bestimmten linguistischen Kategorie. Die folgende Tabelle beinhaltet mögliche Normalisierungsgrößen.

	Korpus A	Korpus B
Token	888169	1192032
Textwörter	768190	1029398
Nomina	183977	246584
Verben	106852	143102
Vollverben	68416	92004
Sätze (Haupt- und Nebensätze)	72386	96879
Nebensätze	19888	26617

Entscheiden Sie, welche Normalisierungsgröße jeweils am geeignetsten ist und rechnen Sie den entsprechenden normalisierten Wert pro 100 Einheiten aus. Runden Sie das Ergebnis auf eine Nachkommastelle.

a)

	Korpus A	Korpus B
absoluter Wert: Wörter mit mindestens einem <e> oder <E>	501001	662326

b)

	Korpus A	Korpus B
absoluter Wert: Fragesätze	736	890

c)

	Korpus A	Korpus B
absoluter Wert: Bewegungs-verben	2549	1931

d)

	Korpus A	Korpus B
absoluter Wert: Subjunktion obwohl	147	202

e)

	Korpus A	Korpus B
absoluter Wert: Attributsätze	8634	11208

f)

	Korpus A	Korpus B
absoluter Wert: modale Adverbiale	34193	46577

3. Berechnen Sie im CQP-Suchinterface (https://hu.berlin/cqp; Login: CQP_Demo, Passwort: TestSuchen) für das Korpus »Akademisches Deutsch« die relativen Häufigkeiten für das Auftreten subordinierender Konjunktionen in den Subkorpora Chemie (Metadatenwert: »chemie«), Physik (Metadatenwert: »psysik«) und Medizin (Metadatenwert: »medizin«). Beantworten Sie die folgende Frage: Sind die Vorkommen in den drei Subkorpora eher ähnlich häufig oder verschieden häufig? Bitte beachten Sie zur Korpussuche mit Metadaten die Informationen in Kap. 3.1.2.27 (insbesondere Szenario 2).

Lösungen

1. a) Sätze (alle Teilsätze)
b) Sätze (alle Teilsätze)
c) Nebensätze oder Attributsätze (als Untergruppe von Nebensätzen)
d) Nomina
e) Tilgbare Schwa-Vorkommen sowie getilgte Schwas
f) Wörter
g) Diskursrelationen
h) Sätze
i) Alle Turn-Übernahmen, inklusive Unterbrechungen

2. a) Normalisierungsgröße: Wörter (Textwörter)
Ergebnis Korpus A: 501001/768190 × 100 ≈ 65,2
Ergebnis Korpus B: 662326/1029398 × 100 ≈ 64,3
b) Normalisierungsgröße: Sätze (Haupt- und Nebensätze)
Ergebnis Korpus A: 736/72386 × 100 ≈ 1,0
Ergebnis Korpus B: 890/96879 × 100 ≈ 0,9

c) Normalisierungsgröße: Vollverben
Ergebnis Korpus A: 2549/68416 × 100 ≈ 3,7
Ergebnis Korpus B: 1931/92004 × 100 ≈ 2,1

d) Normalisierungsgröße: Nebensätze (bester Näherungswert; eigentlich Subjunktionen, ggf. zuzüglich uneingeleiteter Nebensätze)
Ergebnis Korpus A: 147/19888 × 100 ≈ 0,74
Ergebnis Korpus B: 202/26617 × 100 ≈ 0,76

e) Normalisierungsgröße: Nebensätze
Ergebnis Korpus A: 8634/19888 × 100 ≈ 43,4
Ergebnis Korpus B: 11208/26617 × 100 ≈ 42,1

f) Normalisierungsgröße: Vollverben (das ist sinnvoll, weil strukturell jedes Vollverb von einem modalen Adverbial als Modifikator begleitet werden kann; wenn die Anzahl der Adverbiale bekannt wäre, wäre dies auch ein möglicher Normalisierungswert, der jedoch zu einer ganz anderen Aussage führen würde)
Ergebnis Korpus A: 34193/68416 × 100 ≈ 50,0
Ergebnis Korpus B: 46577/92004 × 100 ≈ 50,6

3. Als Normalisierungsgröße sollten am besten Satzbeendungszeichen (STTS: »$.«) und somit abgeschlossene Sätze genommen werden.

- Suchanfrage Chemie: [pos = "KOUS"] ::match.abstract_sachgebiet = "chemie"
  → 615 Treffer
  Normalisierungsgröße ([pos = "\$\."]::match.abstract_sachgebiet = "chemie"): 5265
  Normalisierter Wert ≈0,12
- Suchanfrage Physik: [pos = "KOUS"] ::match.abstract_sachgebiet = "physik"
  → 413 Treffer
  Normalisierungsgröße: ([pos = "\$\."]::match.abstract_sachgebiet = "physik"): 3856
  Normalisierter Wert ≈0,11
- Suchanfrage Chemie: [pos = "KOUS"] ::match.abstract_sachgebiet = "medizin"
  → 6732 Treffer
  Normalisierungsgröße: ([pos = "\$\."]::match.abstract_sachgebiet = "medizin"): 66083
  Normalisierter Wert ≈0,10

Ergebnis: Die normalisierte Häufigkeit von Subjunktionen in den drei Subkorpora ist (mit 12 in »chemie«, 11 »physik« und 10 in »medizin« Subjunktionen pro 100 Sätzen) sehr konstant bzw. vergleichbar.

Arbeitsaufgaben 4.6.1

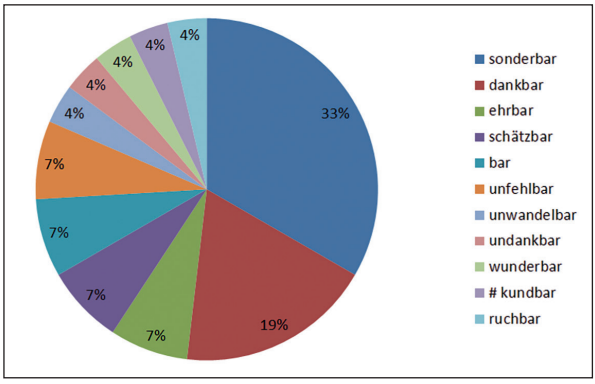
1. Erstellen Sie für das Korpus »Fuerstinnenkorrespondenz1.1« (Zugang: https://hu.berlin/annis-intro) analog zu den oben stehenden Anleitungen ein Variantenprofil, das alle Wortformen (Variable: »tok«) aufzeigt, deren Lemma (Variable: »lemma«) auf *–bar* endet. (Dies ist interessant, weil das Suffix *–bar* als produktiv für das aktuelle Deutsch gilt. Man kann hier sehen, ob dies im 16. – 18. Jh. auch schon so galt.)

2. Erstellen Sie für das Korpus »tiger2« (Zugang: https://hu.berlin/annis-intro) analog zu den oben stehenden Anleitungen ein Variantenprofil, das zeigt, als welche Wortart das Wort *an* besitzen kann. (Die relevanten Variablennamen hierfür sind »lemma« und »pos«.)
3. Erstellen Sie zu der Suchanfrage nach präpositionalem und postnominalem *wegen* in Aufgabe 1 des Kap. 3.1.2.18 eine Übersicht zur Verteilung der Varianten
   a) präpositionales *wegen* mit komplexer Nominalphrase (Artikelwort und/oder Adjektiv) im Vorfeld
   b) präpositionales *wegen* mit Nomen und keinem anderen Wort (kein Artikelwort oder Adjektiv) im Vorfeld
   c) postpositionales *wegen* mit komplexer Nominalphrase (Artikelwort und/oder Adjektiv) im Vorfeld
   d) präpositionales *wegen* mit Nomen und keinem anderen Wort (kein Artikelwort oder Adjektiv) im Vorfeld

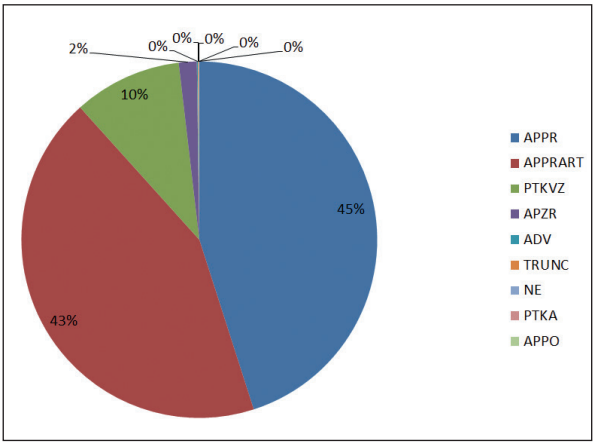
   im DeWaC-1-Korpus des CQP-Webinterfaces der Humboldt-Universität zu Berlin (Zugang über https://hu.berlin/cqp; Nutzername: CQP_Demo, Passwort: TestSuchen).

Lösungen

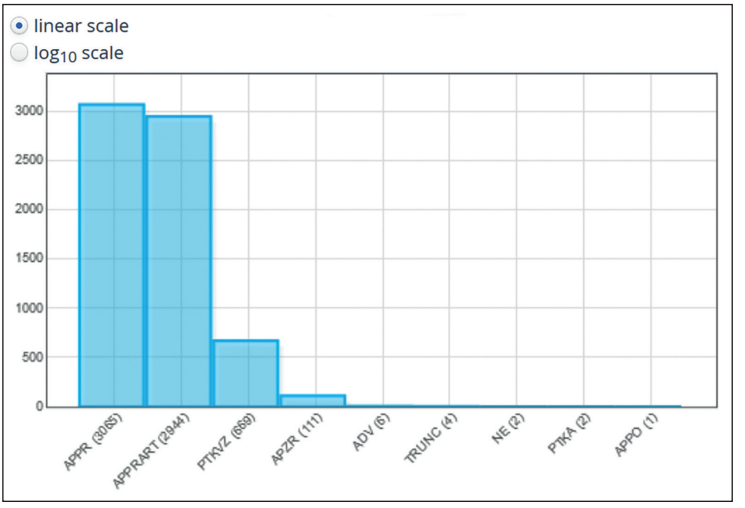
1. Suchanfrage: lemma = /.*bar/
   Hinweis: Schließen Sie nicht-adjektivische Ergebnisse aus, indem Sie die Suche formulieren: lemma = /\#.*/ _ = _ pos = /ADJ./
   Werten Sie die Ergebnisse mit der Funktion »More« > »Frequency Analysis« aus (löschen Sie die Variable »pos«, behalten Sie »lemma«).
   Ergebnis als Kreisdiagramm:



2. Suchanfrage: lemma = "an"
Um die Suchvariable »pos« in die Suche zu integrieren: lemma = "an" _ = _ pos
Werten Sie das Ergebnis mit der Funktion »More« > »Frequency Analysis« nach der Variable »pos« aus (löschen Sie die Variable »lemma«.
Ergebnis als Kreisdiagramm:



Ergebnis als Säulendiagramm (ANNIS-interne Auswertung):

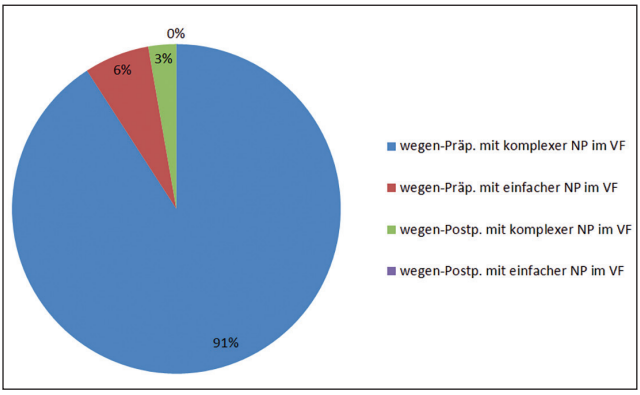


3. Sie müssen zunächst die Suchanfrage für b) formulieren:
[pos = "\$\."] [lemma = "wegen"] [pos = "NN"] [pos = "V.FIN"]
Ergebnis für b): Es gibt im Korpus »DeWaC 1« 140 Vorkommen von *wegen* mit ausschließlich einem Nomen im Vorfeld.

Lösungsweg für a): Suchanfrage:
[pos = "\$\."] [lemma = "wegen"] [pos = "(ART|P.*AT)"]? [pos = "ADJA"]* [pos = "NN"] [pos = "V.FIN"]
Diese Suchanfrage findet auch Fälle von *wegen* mit Artikel und/oder Adjektiven. Hiervon gibt es 2124 Fälle. Die Differenz von 2124 und 140, also 1984, muss die Zahl der komplexen PPn mit *wegen sein*.
Ergebnis für a): 1984

Suchanfrage für d): [pos = "\$\."] [pos = "NN"] [lemma = "wegen"] [pos = "V.FIN"]
Ergebnis für d): keine Treffer (bei den drei gefundenen Treffern handelt es sich um falsche Treffer)

Suchanfrage für c): [pos = "\$\."] [pos = "(ART|P.*AT)"]? [pos = "ADJA"]* [pos = "NN"] [lemma = "wegen"] [pos = "V.FIN"]
Ergebnis für c): 60 Treffer (63, abzüglich der drei falschen Treffer von d))
Gesamtergebnis als Kreisdiagramm:



Arbeitsaufgabe 4.6.2

Verfolgen Sie das Auswertungszenario wie beschrieben weiter und ermitteln Sie anhand der »DeWaC 1«-Korpusdaten vergleichend zu dem Paar *leihen* und *Geld* MI-Werte (mutual information) für *leihen* und *Ohr*, *leihen* und *Bleistift* sowie *schenken* und *Aufmerksamkeit*.

Lösung

- *leihen-Ohr:*
  Adjazentes Auftreten: 19
  Gemeinsames Auftreten im Satz (DeWaC 1): 36
  leihen (Lemma): 1452
  Ohr (Lemma): 10230
  Korpusgröße (Token): 268.849.871
  Erwartete Häufigkeit: ≈0,05

MI (adjazentes Auftreten): ≈343,9
MI (gemeinsames Auftreten im Satz): ≈651,6
Hinweis: Es wurde mit dem nicht-gerundeten Wert für die erwartete Häufigkeit weitergerechnet (im Beispiel im Text ebenso).

- *leihen-Bleistift:*
  Adjazentes Auftreten: 2
  Gemeinsames Auftreten im Satz (DeWaC 1): 2
  leihen (Lemma): 1452
  Bleistift (Lemma): 696
  Korpusgröße (Token): 268.849.871
  Erwartete Häufigkeit: ≈0,004
  MI (adjazentes Auftreten): ≈532, 1
  MI (gemeinsames Auftreten im Satz): ≈532,1
  Hinweis: Es wurde mit dem nicht-gerundeten Wert für die erwartete Häufigkeit weitergerechnet.
- *schenken-Aufmerksamkeit:*
  Adjazentes Auftreten: 442
  Gemeinsames Auftreten im Satz (DeWaC 1): 719
  schenken (Lemma): 10696
  Aufmerksamkeit (Lemma): 10547
  Korpusgröße (Token): 268.849.871
  Erwartete Häufigkeit: ≈0,42
  MI (adjazentes Auftreten): ≈1053,4
  MI (gemeinsames Auftreten im Satz): ≈1713,5
  Hinweis: Es wurde mit dem nicht-gerundeten Wert für die erwartete Häufigkeit weitergerechnet.

Interpretation

- Die Ergebnisse zeigen die relative Vergleichbarkeit der jeweiligen Paare bei konstant bleibender Datenmenge. Betrachtet man die adjazente Stellung der Lexeme, so ergibt sich die folgende Rangfolge: *schenken-Aufmerksamkeit* > *leihen-Bleistift* > *leihen-Ohr* > *leihen Geld*.
- Man sieht, dass sich die Rangfolge ändern kann, wenn man das Gemeinsamauftreten anders definiert: Wenn wir statt adjazentem Auftreten gemeinsame Vorkommen im Satz betrachten, ändert sich die Rangfolge: *leihen-Ohr* ist dann stärker assoziiert als *leihen-Bleistift*.
- Trotz sehr seltenem gemeinsamem Auftreten kann eine Kollokation relativ stark sein, wie man an *leihen-Bleistift* sieht, welches nur zweimal als Kookkurrenz im Korpus auftritt, aber dennoch einen relativ hohen Wert für MI erhält, weil *Bleistift* ein relativ seltenes Wort im Korpus ist.

http://www.springer.com/978-3-476-02643-9

Korpuslinguistik
Eine Einführung
Hirschmann, H.
2019, VIII, 240 S. 57 Abb., 33 Abb. in Farbe., Softcover
ISBN: 978-3-476-02643-9