Een corpus waar alle constructies in gevonden zouden moeten kunnen worden?

Corpusonderzoek met behulp van automatisch gegenereerde syntactische annotatie

Bloem, J.

DOI
10.5117/NEDTAA2020.1.003.BLOE

Publication date
2020

Document Version
Final published version

Published in
Nederlandse Taalkunde

Een corpus waar alle constructies in gevonden zouden moeten kunnen worden?*

Corpusonderzoek met behulp van automatisch gegenereerde syntactische annotatie

Jelke Bloem

NT 25 (1): 39–71
DOI: 10.5117/NEDTAA2020.1.003.BLOE

Abstract

In this contribution, I discuss the use of automatic syntactic annotation in Dutch corpus research, using a case study of five-verb clusters. Large amounts of text can be annotated automatically, but the parser makes mistakes, while correct annotation is very important in linguistic research. How much of a problem is this, and how can we learn about the extent of these parsing mistakes? There are several approaches to evaluating the quality of automatic annotation for specific research questions. I demonstrate these approaches for the case study at hand, which will help us to make claims based on automatically annotated corpus data with greater confidence.

Keywords: corpus linguistics, evaluation, automatic annotation, verb clusters

1. Inleiding

In theorie wordt het leven van taalkundigen steeds makkelijker, omdat steeds meer tekst digitaal beschikbaar is die als data voor taalkundig onderzoek gebruikt kan worden. Toch wordt hier maar beperkt gebruik van

* Ik wil graag het publiek op de Dag van de Nederlandse Zinsbouw (Gent, 21 december 2018) en de reviewer voor *Nederlandse Taalkunde* bedanken voor hun constructieve opmerkingen, evenals de redacteuren van het themanummer voor hun behulpzame redactionele begeleiding.

gemaakt, omdat niet altijd duidelijk is hoe we in deze zee van data de uitingen en constructies kunnen vinden waarin we als taalkundigen geïnteresseerd zijn. Automatische taalkundige annotatiesoftware helpt hierbij, maar het gebruik ervan roept ook twijfels op omdat automatische annotatie onvermijdelijk fouten bevat. In dit artikel zal ik bespreken in hoeverre dat een probleem vormt voor het gebruik van deze data voor taalkundig onderzoek, en hoe de kwaliteit van deze annotatie vanuit taalkundig perspectief beoordeeld kan worden.

Eerder is al gebleken dat de grote hoeveelheid data die automatische annotatie oplevert, wel degelijk iets toe kan voegen aan taalkundig onderzoek (Van Noord & Bouma 2009), ook in vergelijking tot eenzelfde onderzoek uitgevoerd op een kleiner, handmatig geannoteerd tekstcorpus (Bloem, Versloot & Weerman 2014). In dit artikel beoordeel ik of met behulp van deze grotere, automatisch geannoteerde databronnen, voorbeelden van zeldzame constructies uit natuurlijke taal gevonden kunnen worden. Hiervoor ga ik op zoek naar een constructie waar naar mijn weten nog geen natuurlijke taaldata van onderzocht zijn: vijfledige werkwoordsclusters. Hierbij ligt mijn aandacht vooral bij de kwaliteit van de gebruikte automatische annotatie voor het doel van het opzoeken van voorbeelden van deze specifieke constructie, en manieren waarop deze kwaliteit gemeten kan worden.

Door dit zeer infrequente fenomeen als voorbeeld te nemen, kunnen we beoordelen of het mogelijk is om op basis van automatische annotatie daadwerkelijk voorbeelden te vinden van een constructie die niet in een handmatig geannoteerd corpus te vinden is. Dit lijkt op te gaan voor vijfledige werkwoordsclusters — ze zijn in ieder geval niet aangetroffen door Augustinus (2015) in haar uitgebreide corpusonderzoek naar werkwoordsclusters, waarvoor ze gebruik maakte van het handmatig geannoteerde Corpus Gesproken Nederlands en Lassy Klein. Zij merkt hierbij op dat "the data indicate that they are avoided in actual language use" Augustinus (2015: 126).

Toch wordt aan de grammaticaliteit van deze constructie niet getwijfeld. Ze wordt besproken in referentiewerken als de Algemene Nederlandse Spraakkunst (Haeseryn et al. 1997), waarbij (geconstrueerde?) voorbeelden gegeven worden, en ook genoemd door taalkundigen als Stroop (1970). Ik voeg ter illustratie nog een paar nieuwe voorbeelden toe aan dit corpus van geconstrueerde voorbeelden:

(1) Is er een corpus waarin alle constructies **gevonden zouden moeten kunnen worden?**

(2) Is er een corpus waarin alle constructies **zouden moeten kunnen worden gevonden?**
(3) Is er een corpus waarin alle constructies **zouden moeten kunnen gevonden worden?**

In het Nederlands, net als in andere Germaanse talen, drukken we eigenschappen als tijd en aspect vaak uit met behulp van hulpwerkwoorden, die dan in het Nederlands samen aan het eind van de zin geplaatst worden, behalve het finiete werkwoord in de hoofdzin. Hierdoor kunnen, vooral in de bijzin, grote groepen werkwoorden gebruikt worden, die ook syntactisch van elkaar afhankelijk zijn. Dit zijn werkwoordsclusters, en in bovenstaande geconstrueerde voorbeelden bevatten deze vijf werkwoorden: vijfledige werkwoordsclusters. Opvallend in het Nederlands is dat verschillende woordvolgorden gebruikt worden: bij twee werkwoorden kunnen, vooral wanneer er sprake is van een participium als hoofdwerkwoord, beide mogelijke volgorden gebruikt worden: het hulpwerkwoord ervoor of het hulpwerkwoord erna.

(4) Ik zei dat ik het **gehoord heb**.
(5) Ik zei dat ik het **heb gehoord**.

Wanneer het hulpwerkwoord voorop geplaatst wordt, zoals in voorbeeld 5, wordt dit de 1-2-volgorde genoemd, omdat het syntactisch hoogste werkwoord, het hulpwerkwoord (aangeduid met 1) eerst komt. Ook wordt de term *rode volgorde* gebruikt. Voorbeeld 4 is dan de 2-1 volgorde, of de *groene volgorde*.

Ook bij vijfledige clusters lijkt sprake te zijn van variatie: voorbeelden 1, 2 en 3 zijn alle denkbaar. Met de gebruikelijke notatie kunnen we deze aanduiden als de 5-1-2-3-4-volgorde (door Stroop 1970 een tangconstructie genoemd), de 1-2-3-4-5-volgorde en de 1-2-3-5-4-volgorde. Dit zijn allemaal 'stijgende' volgorden, waarbij alleen de positie van het voltooid deelwoord varieert. In sommige taalvariëteiten, vooral in het noorden, zijn dalende volgorden mogelijk. Bloemhoff (1979) geeft voorbeelden met een 4-3-2-1-volgorde uit het Stellingwerfs.

Hoewel ik met dit artikel vooral een methodologisch punt probeer te maken, is het wel degelijk ook vanuit theoretisch perspectief interessant om specifiek naar vijfledige werkwoordsclusters te kijken. In tweeledige werkwoordsclusters is veel variatie mogelijk, maar hoe langer de clusters worden, hoe minder variatie er optreedt, doordat het Nederlands slechts een beperkt repertoire aan groepsvormende werkwoorden heeft, die door

hun betekenis niet eindeloos en in elke positie gecombineerd kunnen worden. De variatie die nog wel mogelijk is, zal patronen volgen die ook in kleinere werkwoordsclusters zichtbaar zijn. Over vijfledige werkwoordsclusters zijn, voor zover ik weet, geen specifieke hypothesen geformuleerd, maar het is wel mogelijk om theorieën over de variatiemogelijkheden in kleinere werkwoordsclusters te toetsen aan data over vijfledige clusters. Uit het werk van Stroop (2009) zouden we bijvoorbeeld kunnen extrapoleren dat er in Vlaanderen een voorkeur zal zijn voor de 1-2-3-5-4-volgorde, en in Nederland een voorkeur voor de 5-1-2-3-4-volgorde, maar om dit te onderzoeken is een corpus nodig waarin metadata over het land van publicatie beschikbaar is. Volgens De Schutter (2012) zullen volgorden die een jambisch ritmisch patroon hebben, de voorkeur krijgen. Ook zouden we op basis van Bloem (2016b) kunnen extrapoleren dat een volgorde waarin het hoofdwerkwoord voorop staat, een hogere verwerkingscomplexiteit heeft en dus minder vaak gebruikt zal worden, of in minder complexe contexten gebruikt zal worden. Om dat te beoordelen zijn echter een behoorlijk aantal gevallen nodig, in verschillende soorten contexten. Nog meer factoren die mogelijk een rol spelen worden besproken in het overzicht van Coussé, Arfs & De Sutter (2008).

Om deze constructie te bestuderen, hebben we een syntactisch geannoteerd corpus nodig, ook wel treebank genoemd. Het is niet voldoende om alleen naar een reeks werkwoorden te zoeken, want dat maakt nog geen werkwoordscluster – om een werkwoordscluster te zijn, moeten de werkwoorden de syntactische structuur van een werkwoordscluster hebben, met een hoofdwerkwoord en een aantal hulpwerkwoorden of modale werkwoorden. Reeksen werkwoorden waarbij de werkwoorden uit twee verschillende bijzinnen komen, horen er bijvoorbeeld niet bij. Omdat het om een zeldzame constructie gaat, zal de treebank groot moeten zijn om een voldoende groot aantal gevallen te kunnen vinden. Daarom komen alleen automatisch geannoteerde treebanks in aanmerking.

Natuurlijk zijn er ook andere manieren om data te verzamelen over dit soort zeer infrequente constructies, bijvoorbeeld door middel van enquêtes waarin moedertaalsprekenden zinnen met vijfledige werkwoordsclusters te zien krijgen en hun gevraagd wordt of ze deze zinnen acceptabel vinden. Maar dit levert geen natuurlijke taaldata op, waardoor niet alle relevante factoren bestudeerd kunnen worden en waarbij expliciete taalkennis of vooroordelen het oordeel kunnen beïnvloeden (Gibson & Fedorenko 2013; Pullum 2007).

1.1 Automatisch gegenereerde syntactische annotatie

Treebanks zijn tekstcorpora die verrijkt zijn met syntaxbomen, waardoor deze doorzoekbaar worden op syntactisch en morfologisch niveau. Met een treebank kan gezocht worden naar een bepaalde syntactische constructie, of bepaalde morfologische eigenschappen, en daarmee een lijst van voorkomens van die constructie verkregen worden die veel makkelijker te bestuderen is dan een volledige tekst. Dankzij de computationele taalkunde kunnen deze syntaxbomen ook automatisch door een parser toegevoegd worden aan een corpus, waardoor veel grotere hoeveelheden tekst geannoteerd kunnen worden dan voorheen. Hierdoor zijn grote treebanks beschikbaar gekomen, die veel groter zijn dan handmatig geannoteerde treebanks.[1] Dit levert voor taalkundigen verschillende voordelen op: er zijn natuurlijk grotere steekproeven en meer voorbeeldzinnen uit te halen, maar ze kunnen ook ingezet worden om zeldzamere constructies, die in een kleiner corpus niet te vinden zijn, te onderzoeken. Ook kan nauwkeuriger geteld worden: meer data levert nauwkeuriger berekende collocaties of woordkansen op, en bij alternanties tussen constructies kan ook de kans op een bepaalde constructie, gegeven bepaalde contextuele factoren, nauwkeuriger ingeschat worden.

Deze voordelen hebben ook een keerzijde: automatisch toegevoegde annotatie is minder accuraat. Handmatig geannoteerde corpora bevatten natuurlijk ook annotatiefouten, maar er is een duidelijk verschil.[2] Een deel van de fouten die de parser maakt zullen systematisch zijn, waar het bij bepaalde soorten constructies veel vaker fout gaat dan bij andere. Het kan ook gebeuren dat de parser met bepaalde constructies helemaal niet om kan gaan, waardoor deze niet vindbaar zullen zijn in het corpus. Dit is natuurlijk belangrijk om te weten voor een taalkundige die een specifieke constructie wil onderzoeken. Parsers worden natuurlijk geëvalueerd, en vaak is het goed mogelijk om de algemene accuratesse van een parser of van een specifiek corpus te vinden. Echter, hiermee is nog niet duidelijk wat de accuratesse van het automatisch annoteren van een specifieke constructie is, terwijl dat juist voor veel taalkundig onderzoek het belangrijkste gegeven is.

1 Odijk (dit nummer) hanteert een nader terminologisch onderscheid tussen handmatig gecontroleerde 'treebanks' en niet handmatig gecontroleerde 'parsebanks'.

2 Een goede illustratie hiervan zijn de verhoudingen van correct geanalyseerde zinnen van het handmatig geannoteerde Lassy Klein-corpus en het automatisch geannoteerde Lassy Groot: respectievelijk 97.8% en 78.4% (van Noord et al. 2013).

1.2 Zoeken naar specifieke constructies

Wanneer een taalkundige een specifieke constructie aan het onderzoeken is in een corpus, is zij eigenlijk aan het zoeken. Daarom is het nuttig om corpusonderzoek als een zoektaak te beschouwen: voor een zo representatief mogelijke steekproef moeten alle voorkomens van de doelconstructie uit een grotere databron gehaald worden. Bij zoeken horen de maten *precision* en *recall*. *Precision*, of precisie, is bij corpusonderzoek het percentage resultaten van een corpuszoekopdracht die gevallen van de doelconstructie zijn. Het zijn dus alle resultaten die geen foutpositieve resultaten zijn. *Recall*, of opbrengst, is het percentage van alle gevallen van de doelconstructie in het corpus die daadwerkelijk gevonden worden door de zoekopdracht. Het zijn dus alle resultaten die geen foutnegatieve gevallen zijn.

Imperfecte *precision* of *recall* kan meerdere oorzaken hebben. Het kan zo zijn dat de onderzoeker de doelconstructie niet helemaal goed gedefinieerd heeft in de zoekopdracht, en daardoor ook resultaten krijgt die geen gevallen van de doelconstructie zijn: een zoekfout. Ook kan het zijn dat bepaalde gevallen van de doelconstructie niet correct geannoteerd zijn, en daarom niet gevonden worden: een annotatiefout. Ten slotte kan het nog zo zijn dat het annotatieformaat van het corpus de doelconstructie niet kan beschrijven, en hij daardoor niet gevonden kan worden. Een voorbeeld hiervan is wanneer de onderzoeker naar connectieven zoals *omdat* wil zoeken: in de meeste annotatieschema's die treebanks hanteren is hier geen aparte categorie voor, dus kan men ook niet naar connectieven zoeken, behalve door een lijst van alle mogelijke connectieven af te gaan. Deze categorie fouten is zeker belangrijk voor taalkundigen, maar zal ik hier verder buiten beschouwing laten. Er is vaak weinig aan te doen wanneer dit optreedt, behalve een ander corpus proberen te vinden waar deze eigenschap wel in geannoteerd is.

Ik begin in sectie 2 met een bespreking van enkele studies waarin al gebruik gemaakt is van automatisch geannoteerde corpora. In sectie 3 bespreek ik hoe in deze artikelen naar de kwaliteit van deze annotatie gekeken wordt en welke benaderingen hiervoor voorgesteld zijn. In sectie 4 ga ik in op de data en evaluatiemethoden die ik voor dit onderzoek gebruik, en sectie 5 bespreekt de resultaten van mijn voorbeeldstudie, met een discussie in sectie 6.

2. Onderzoek met automatisch geannoteerde corpora

Automatisch geannoteerde treebanks zijn inmiddels al een tijd beschikbaar (Van Noord 2009),[3] en ze worden ook gebruikt voor taalkundig onderzoek, maar het gebruik ervan lijkt nog niet echt wijdverbreid. In deze paragraaf zal ik een paar voorbeelden noemen van dit soort treebanks en taalkundige studies die er gebruik van hebben gemaakt. Dit is natuurlijk geen volledig overzicht. Verder moet opgemerkt worden dat automatische annotatie in de computationele taalkunde veel meer gebruikt wordt als verwerkingsstap voor verdere taken, bijvoorbeeld voor het detecteren van coreferenten (Clark & Manning 2016), maar dat zal ik hier verder buiten beschouwing laten.

Voor het Nederlands is Lassy Groot beschikbaar, een treebank van rond de 700 miljoen woorden (Van Noord et al. 2013) Grotere web-corpora zijn ook beschikbaar, zoals NLCOW14 (Schäfer & Bildhauer 2012), maar vaak zonder syntactische annotatie – dit zijn dus geen treebanks, al zou het met enige moeite mogelijk zijn om zelf met een parser syntaxbomen toe te voegen. Voor het Duits is er onder andere de TüPP-D/Z (Müller 2004), een automatisch geannoteerde treebank met 200 miljoen woorden. Voor het Engels zijn meer en grotere corpora beschikbaar, zoals de syntactisch geannoteerde versie van het Google Books n-gram corpus (Lin et al. 2012), en het Gigaword v5 corpus (Napoles, Gormley & Van Durme 2012) met 4 miljard woorden. Voor elke taalvariëteit of taaldomein waar een parser voor beschikbaar is, kan een nieuwe treebank gemaakt worden, en zulke parsers bestaan voor veel van de grotere talen (maar bijvoorbeeld niet voor het Fries). Er is recent ook veel werk verricht om deze corpora en de bijbehorende zoektechnologie toegankelijker te maken voor taalkundigen, die geen technische achtergrond hebben of niet kunnen programmeren. Voorbeelden hiervan zijn example-based querying, waar op basis van een voorbeeldzin naar zinnen met dezelfde syntactische structuur gezocht kan worden (Augustinus, Vandeghinste & Van Eynde 2012), en systemen waarbij men eigen teksten kan uploaden om automatisch te laten annoteren, zoals inmiddels mogelijk is in PaQu (*Parse and Query*, Odijk et al. 2017).

Om te zien hoe deze treebanks toegepast kunnen worden, kunnen we inmiddels ook al een aantal jaren terug kijken. Een aantal toepassingen van de Lassy Groot treebank voor het Nederlands worden opgesomd door

3 In 2005 werd nog geschreven dat ze niet bestaan (Kakkonen 2005), al is het TüPP-D/Z corpus een ouder voorbeeld (Müller 2004).

Van Noord en Bouma (2009), die laten zien waarom zo'n groot corpus in deze gevallen nuttig was. Al tijdens de ontwikkeling van de Alpino-parser voor het Nederlands (Van Noord et al. 2006), die gebruikt is om het Lassy Groot-corpus te annoteren, bleek dat met dit type corpusonderzoek aannames over grammaticaliteit empirisch getoetst kunnen worden. Van der Beek, Bouma, en Van Noord (2002) beschrijven de Alpino-grammatica, en gaan met name in op comparatieven. Uit voetnoot 8 (Van der Beek, Bouma, & Van Noord 2002: 364) blijkt dat hun grammaticaregels voor extrapositie vanuit een getopicaliseerde constituent als te breed werden gezien door een beoordelaar: deze beoordelaar claimde dat dit in het geval van extrapositie van bepalingen van vergelijking ongrammaticaal was. Gebruik makend van de boomstructuren die de Alpino-parser oplevert, werden echter wel degelijk voorbeelden van dit soort extrapositie gevonden, en op basis daarvan concluderen de auteurs dat deze extrapositie in bepaalde contexten dus grammaticaal zijn.

Een ander voorbeeld, dat betrekking heeft op een ander deelgebied van de taalkunde is de studie van Bastiaanse & Bouma (2007), die de structuren uit een automatisch geannoteerde treebank inzetten als maat van talige complexiteit. Ze laten zien dat patiënten met afasie van Broca meer moeite hebben met constructies die volgens deze maat een hogere complexiteit hebben, en dat het niet slechts om een frequentie-effect gaat. Bij dit type onderzoek is het voordeel van de grote hoeveelheid automatisch geannoteerde zinnen dat er grotere steekproeven genomen kunnen worden. Voor dit onderzoek waren zinnen met specifiek geselecteerde, op frequentie gebalanceerde werkwoorden vereist, waardoor niet zomaar elke zin in aanmerking kwam. Met een grotere databron om uit te putten, kon de frequentie van het gebruik van deze werkwoorden in bepaalde contexten accuraat geschat worden. Iets soortgelijks is te zien in de studie van Bouma & Spenader (2008), waarin de associatie van *zich* en *zichzelf* met de werkwoorden waarmee ze gebruikt worden bestudeerd wordt. Hiermee wordt gezocht naar relatief zeldzame constructies en de woorden die daarmee verband houden, waarvan in grotere treebanks (die alleen automatisch geannoteerd kunnen worden) meer relevante voorbeelden gevonden zullen worden.

Het is zelfs mogelijk gebleken om kindertaaldata te onderzoeken met behulp van automatische annotatie. Odijk (2015) gebruikte de Alpino-parser om syntactische annotatie aan het CHILDES-corpus toe te voegen, ondanks dat deze parser geen specifieke aanpassingen heeft om met kindertaaldata om te gaan. Hiermee bleek het goed mogelijk om onderzoek te doen naar de verwerving van de bijwoorden *heel*, *erg* en *zeer* (zie ook

Odijk, dit nummer). Ondanks dat de parser fouten maakte, vooral in het geval van het infrequente en ambigue *zeer*, bleken de resultaten toch goed bruikbaar, dankzij de focus op een specifiek fenomeen waarbij de syntactische structuren klein blijven. Uit het onderzoek bleek dat zelfs in kindgerichte spraak, *heel* al andere distributionele eigenschappen heeft dan *erg*, op basis waarvan kinderen het patroon zouden kunnen oppikken dat *heel* geen werkwoorden modificeert.

Automatisch geannoteerde corpora zijn ook gebruikt om bewijs te vinden voor zeldzame constructies vanwege hun grootte. Een veelgehoorde kritiek op de corpustaalkunde is dat lang niet alles wat grammaticaal is, ook in een corpus te vinden is, en dat zeldzame (combinaties van) constructies er niet in te vinden zijn (Pullum 2017). Discussies over zeldzame constructies worden vaak slechts onderbouwd met grammaticaliteitsoordelen, ondanks dat hier veel op aan te merken valt (Gibson & Fedorenko 2013). Bouma (2017) laat zien dat automatische annotatie meer duidelijkheid kan verschaffen over het daadwerkelijk gebruik van lange-afstandsafhankelijkheden. Ondanks dat ze veel besproken worden in de theoretische literatuur, zijn ze zodanig zeldzaam dat niet duidelijk is in hoeverre ze daadwerkelijk gebruikt worden. In het Lassy Groot corpus vindt Bouma (2017) genoeg voorbeelden van specifieke soorten lange-afstandsafhankelijkheden om theoretische claims uit de literatuur over deze constructies te testen.

Dankzij automatische parsers kan dit type onderzoek ook voor meerdere talen uitgevoerd worden. Recent hebben Blasi et al. (2019) onderzocht hoe vaak diep ingebedde bijzinnen nu eigenlijk gebruikt worden, iets wat in de theoretische taalkunde vaak als fundamentele eigenschap van menselijke taal beschouwd wordt, maar ook erg zeldzaam lijkt te zijn. Met behulp van parsers die Universal Dependencies (Nivre et al. 2016) produceren, een universeel syntactisch annotatieformaat dat voor veel talen beschikbaar is, konden ze dit voor 17 talen tegelijk onderzoeken

Ook onderzoek naar combinaties van woorden of constructies leent zich goed voor het gebruik van automatisch geannoteerde corpora. Om bijvoorbeeld onderzoek te doen naar werkwoordsclusters waarbij extrapositie van het voorzetselvoorwerp optreedt, wordt het onderzoeksdomein niet alleen beperkt tot zinnen met werkwoordsclusters, maar ook nog tot slechts die zinnen met werkwoordsclusters waar een voorzetselvoorwerp gebruikt wordt, en waarbij dat voorzetselvoorwerp na het werkwoordscluster komt.[4]

4 Dit is een hypothetisch voorbeeld, dit onderzoek is nog niet uitgevoerd maar zou wel interessant zijn, zoals voorgesteld in Bloem, Versloot & Weerman (2017).

Hoe meer elementen er zijn in de combinatie die onderzocht wordt, hoe groter het corpus dat nodig is om nog een behoorlijk aantal voorbeelden te vinden waarmee statistisch interessante resultaten gemodelleerd kunnen worden. Onderzoek naar de volgorde binnen werkwoordsclusters is ook al met automatisch geannoteerde corpora uitgevoerd. Voor het Duits is op basis van de TüPP-D/Z treebank een onderzoek uitgevoerd naar vooropplaatsing van het hulpwerkwoord in drieledige werkwoordsclusters (Hinrichs & Beck 2013), een vrij zeldzame combinatie van factoren. Hiervoor was een zeer groot corpus nodig. Hinrichs en Beck rapporteren welke werkwoorden gebruikt worden in deze constructie en vergelijken de data uit de treebank met (nog veel zeldzamere) gevallen uit kleinere diachrone corpora.

Ook bij onderzoek naar taalvariatie lenen automatisch geannoteerde corpora zich goed voor het inschatten van de kans op een bepaalde variant. Dit is bijvoorbeeld het geval bij alternanties, waar vaak een aantal factoren van invloed zijn die de kans verhogen of verkleinen dat een bepaalde variant gebruikt wordt. Voor onderzoek naar alternanties worden doorgaans forse corpora gebruikt om een constructie in allerlei verschillende contexten te kunnen vinden en deze contexten als factoren te kunnen bestuderen. Dit soort probabilistische effecten, bijvoorbeeld van woordfrequentie of de complexiteit van een constituent, kunnen accurater ingeschat worden wanneer daar meer data voor beschikbaar zijn. Voor tweeledige werkwoordsclusters zijn een groot aantal factoren onderzocht die van invloed kunnen zijn op de volgorde van de twee werkwoorden (Coussé, Arfs, en De Sutter 2008), zowel met behulp van handmatig geannoteerde data (De Sutter 2005) als met automatisch geannoteerde data (Bloem, Versloot & Weerman 2014). Waar de eerstgenoemde studie op basis van 2390 werkwoordsclusters werd uitgevoerd, werden uit het automatisch geannoteerde corpus 411623 gevallen gehaald. Het annotatieschema van de automatische annotatie bleek hierbij beperkingen op te leveren, maar toch konden de resultaten van de studie op basis van handmatige annotatie grotendeels gerepliceerd worden, en ook uitgebreid doordat meer condities en varianten van de werkwoordsclusterconstructie automatisch uit het corpus gehaald konden worden.

3. De kwaliteit van automatische annotatie

Een veelgenoemde reden om sceptisch te zijn over automatisch geannoteerde corpora is dat de kwaliteit van de annotatie slechter is, omdat de annotatie niet door mensenhanden geproduceerd is of zelfs maar geredigeerd

is. Hierbij wil ik de vraag stellen of we wel weten hoe goed of hoe slecht die kwaliteit dan precies is. Dit is lang niet altijd duidelijk, vooral voor het doel van een specifiek taalkundig onderzoek. Hieronder ga ik in op hoe de kwaliteit van automatische annotatie gemeten kan worden, en geëvalueerd wordt in studies zoals de bovengenoemde.

Om de kwaliteit van automatische annotatie te meten, kunnen we beginnen bij iets wat wel duidelijk te meten is: de accuratesse van de parser waarmee de treebank geannoteerd is. Dit kan dan ook gezien worden als een kwaliteitsmaat voor de treebank in zijn geheel. Om de accuratesse van de parser te meten op een bepaalde dataset, moet een deel van deze dataset handmatig geannoteerd worden, waarna de output van de parser vergeleken kan worden met de output van de menselijke annotators. Hierbij wordt dan een score berekend, zoals de Attachment Score, het percentage woorden die het correcte syntactische hoofd toegewezen gekregen hebben door de parser. Eventueel wordt hierbij meegerekend of ook de juiste relatiesoort benoemd is. De Alpino-parser voor het Nederlands is op deze wijze geëvalueerd, en de bereikte score (86,52%) kan dan gezien worden als maat van de kwaliteit van de annotatie. Hierbij is deze evaluatie ook nog voor specifieke domeinen uitgevoerd, bijvoorbeeld alleen zinnen uit Wikipedia (88,38%) of alleen zinnen uit boeken (92,86%) (Van Noord 2009). Studies waarbij gebruik gemaakt van een corpus dat door Alpino geannoteerd is, rapporteren deze percentages dan soms ook.

Voor taalkundig onderzoek zijn we meestal echter geïnteresseerd in specifieke constructies, en niet zo vaak in volledige tekstdomeinen. Hierbij is het gebruik van dit soort algemene scores een probleem, want fouten zullen niet evenredig verdeeld zijn over verschillende soorten constructies. Relaties tussen een lidwoord en het bijbehorende zelfstandig naamwoord zullen vaker goed geannoteerd worden dan lange-afstandsafhankelijkheden, en die lokale lidwoordverbanden zullen ook nog eens veel frequenter zijn. Een parser zou dan misschien wel een algemene score van 95% kunnen behalen, maar als 0% van de lange-afstandsafhankelijkheden goed gaan, zal het automatisch geannoteerde corpus geen goede bron zijn voor een onderzoek daarnaar. Met andere woorden, parseerfouten zijn systematisch.

Het systeem achter de fouten kan al deels voorspeld worden door te kijken naar hoe een parser werkt. Tegenwoordig zijn bijna alle parsers gebaseerd op statistische leermethoden. Een logisch gevolg hiervan is dat parsers meer fouten maken bij zeldzamere fenomenen, waar de parser minder voorbeelden van heeft gezien bij het trainen. Ook zien we dat parsers slechter werken op langere zinnen, wat voor de Alpino-parser te zien is

in Van Noord et al. (2006: 11, Fig. 5). Bij ambiguïteit worden ook meer fouten gemaakt, wat al sinds de tijd van regelgebaseerde systemen een probleem was. Verder geldt over het algemeen dat er meer fouten zullen optreden wanneer de geannoteerde data meer afwijkt van de data waarop de parser getraind is. Dit geldt bijvoorbeeld voor de kindertaaldata die Odijk (2015) bestudeerde, geannoteerd met een parser die op 'volwassen' tekst getraind is. Hierdoor verschilt de kwaliteit van automatische annotatie afhankelijk van welk fenomeen bestudeerd wordt.

Helaas zijn zeldzame fenomenen nu net vaak ook het interessantst voor taalkundigen, wat op een potentieel probleem wijst. Ook zinslengte wordt niet zelden gebruikt als relevante factor in corpusstudies. Deze patronen in de gemaakte fouten zorgen ervoor dat bepaalde constructies vaker fout geparseerd worden dan andere, en dit kan ongewenste vertekeningen in de onderzoeksresultaten opleveren. Hierom lijkt het mij wenselijk voor een taalkundige die gebruik maakt van dit soort corpora om aan constructie-specifieke evaluatie te doen, waarmee de kwaliteit van de gebruikte data beter ingeschat kan worden. Om onderzoek te doen naar werkwoordsclusters is het bijvoorbeeld goed om te weten wat de accuratesse is van relaties tussen werkwoorden, en zouden we de score op relaties tussen lidwoorden en hun zelfstandig naamwoord liever niet meetellen.

Tabel 1 Overzicht van manieren om de kwaliteit van gevonden corpusdata te evalueren, aangepast uit Bloem (2016a)

Benadering	Nadelen	Voordelen
Handmatige evaluatie van de zoekresultaten	Geen recall, kost tijd	Maat van precisie
Terugvallen op eenvoudigere annotatielaag	POS-tagfouten niet zichtbaar	Maat van recall
Naar specifieke gevallen zoeken	Moeilijk te generaliseren	Maat van recall
Handmatige evaluatie van tekst	Kost heel veel tijd	Precisie & recall

Sommige onderzoeken houden hier al rekening mee door de automatisch geannoteerde zinnen te vergelijken met een handmatig geannoteerd referentiecorpus. Dit wordt bijvoorbeeld gedaan door Odijk (2015), die als maat van kwaliteit de accuratesse van de parser op kindertaal berekent door een handmatig geannoteerd deel van het corpus te parsen, en te berekenen in hoeverre de twee syntaxbomen overeen komen. Zo'n handmatig geannoteerd deelcorpus is echter niet altijd beschikbaar, en kost veel tijd en moeite om te maken. Bovendien bevatte de handmatige annotatie in dit geval ook fouten. Door Bloem, Versloot, en Weerman (2014) wordt een semi-automatische aanpak gehanteerd: in hun onderzoek naar werkwoordsclusters controleren ze handmatig een deel van de opgeleverde resultaten. Hiermee

wordt echter niet duidelijk of er ook werkwoordsclusters in de resultaten ontbraken. Elk werkwoordscluster dat door de parser de verkeerde syntactische structuur toegewezen heeft gekregen, zal niet gevonden worden door een zoekopdracht naar die structuur, en zit dus niet in de steekproef. In andere artikelen wordt helemaal geen aandacht besteed aan constructiespecifieke evaluatie, waardoor het niet duidelijk wordt of de data waarop het onderzoek is uitgevoerd, representatief is voor de constructie.

3.1 Constructie-specifieke evaluatie

Er zijn (tenminste) vier mogelijke benaderingen om een constructiespecifieke evaluatie uit te voeren met het doel om inzicht te krijgen in de annotatiekwaliteit van de data die gebruikt wordt om een bepaalde taalkundige onderzoeksvraag te beantwoorden (Bloem 2016a). Hierbij wordt op verschillende manieren naar de doelconstructie in het corpus gezocht, gebruik makend van verschillende informatielagen in het corpus. Tabel 1 geeft een overzicht hiervan. De aannames van deze verschillende benaderingen zijn dat het gebruikte corpus automatisch geannoteerd is, maar dat de onderzoeker geen directe toegang heeft tot de annotatiesoftware om die te analyseren.

De eerste manier is de net genoemde, en misschien de meest vanzelfsprekende: handmatige evaluatie van de zoekresultaten. Hierbij wordt de syntactische structuur van de doelconstructie geformaliseerd in een zoekopdracht, en de resultaten van die zoekopdracht worden handmatig door de onderzoeker, of door een andere expert, beoordeeld. Vaak zal het moeten gaan om een steekproef, omdat een groot corpus veel resultaten oplevert. Elk teruggegeven resultaat in de steekproef dat overeenkomt met de zoekopdracht maar geen geval van de doelconstructie is, wordt als een fout gemarkeerd. Een foutenpercentage kan dan berekend worden, en dit percentage kan gezien worden als een maat van *precision*. Hiermee weten we echter niet hoeveel foutnegatieve resultaten er waren, dus constructies die wel gevonden hadden moeten worden, maar niet in de resultaten verschenen. Het is dus geen maat van *recall*. Dit kunnen bijvoorbeeld gevallen zijn waarin de automatische annotatie niet klopt. Als sommige werkwoorden door de parser zijn aangezien voor bijvoeglijke naamwoorden, dan zullen deze gevallen niet verschijnen in een zoekopdracht naar werkwoorden. Gemiste gevallen hoeven echter niet aan de annotatie te liggen, en kunnen ook komen doordat de onderzoeker de constructie niet goed geformaliseerd heeft in termen van de beschikbare annotatie en daardoor een onnauwkeurige zoekvraag heeft gesteld. Een voorbeeld hiervan is wanneer een zoekopdracht naar tweeledige werkwoordsclusters geen rekening

houdt met de mogelijkheid dat het hoofdwerkwoord een *te*-infinitief is. Aangezien *te*-infinitieven een andere syntactische structuur hebben in het annotatieschema dat Alpino hanteert, moet die mogelijkheid toegelaten worden in een zoekopdracht.

De tweede benadering is om terug te vallen op een eenvoudigere annotatielaag, bijvoorbeeld door alleen te zoeken op lexicale eigenschappen als woordklassen, zonder beperkingen op de syntactische structuur. Het idee hierachter is dat grotere structuren vaker fout geparseerd worden bij automatische annotatie, dus de eenvoudigere annotatielaag van woordklassen zou betere resultaten moeten opleveren. In het geval van werkwoordsclusters kan het bijvoorbeeld gebeuren dat een werkwoord door de parser aan de verkeerde bijzin is gekoppeld, waardoor het geen syntactisch cluster meer vormt met de andere werkwoorden. Wanneer men syntactisch naar een werkwoordscluster zoekt zal dit geval niet verschijnen, maar wanneer men alleen naar een reeks werkwoorden zoekt wel. Bij deze benadering wordt dus een eenvoudigere zoekopdracht ingevoerd: we zoeken alleen een reeks werkwoorden die achter elkaar staan. Dit heeft dan wel weer als nadeel dat ook werkwoorden die geen cluster vormen maar om andere redenen naast elkaar staan, ook in de resultatenlijst verschijnen. Verder zullen op deze manier gevallen van clusterinterruptie, waarbij een niet-werkwoordelijk element in een syntactisch werkwoordscluster staat, buiten de boot vallen. Het voordeel van deze aanpak is dat dit een grotere resultatenlijst zal opleveren dan de eerste benadering, die ook weer niet onhandelbaar groot is. De resultaten van de tweede benadering en de eerste benadering kunnen dan vergeleken worden. Alles wat wel met de tweede benadering gevonden wordt, maar niet met de eerste, kan dan handmatig gecontroleerd worden, en zo worden mogelijk nog gevallen gevonden die ook in de eerste lijst hadden moeten staan. Hiermee wordt het *recall*-probleem van de eerste methode ondervangen.

De derde benadering is om naar specifieke gevallen van de doelconstructie te zoeken, bijvoorbeeld door naar *hebben gedaan* te zoeken bij een onderzoek naar tweeledige werkwoordsclusters. Dit is een *string search* of *keyword search*, simpelweg naar woordenreeksen zoeken. Deze benadering is helemaal niet afhankelijk van annotatie en dus nog eenvoudiger. Daarom kunnen annotatiefouten op deze manier geen rol spelen, ook niet op het niveau van lexicale informatie. Wel kunnen typfouten, transcriptiefouten, spelfouten en dergelijke alsnog roet in het eten gooien. Het nadeel is natuurlijk dat er geen generalisatie plaats kan vinden over verschillende woorden die in een constructie gebruikt kunnen worden, en het zal niet mogelijk zijn om elke variant op deze manier op te sporen. De onderzoeker

moet daarom een representatief voorbeeld kiezen, of een paar, waarmee geëvalueerd kan worden. Net als bij de vorige benadering kunnen deze gevallen, gevonden met *string search*, dan vergeleken worden met de lijst van gevallen die gevonden wordt met syntactisch zoeken. Als een geval in de resultaten van de string search verschijnt, maar niet in de resultaten van de syntactische zoekopdracht, is er waarschijnlijk sprake van een annotatiefout. Ook met deze benadering kunnen dus gevallen gevonden worden die niet met behulp van de syntactische annotatie gevonden worden – dit is dus ook een manier om *recall* te evalueren, al is het alleen voor specifieke gevallen (*types*) van een constructie.

De vierde benadering, nog altijd door sommigen als de beste manier aangeprezen, is om (delen van) de tekst handmatig door te lezen en te inspecteren. Hiermee kunnen inderdaad zowel *precision* als *recall* beoordeeld worden, maar elk voordeel van automatische annotatie verdwijnt ook door de enorme hoeveelheid tijd die erin zou gaan zitten. Vooral met een zeldzame constructie als vijfledige werkwoordsclusters is deze benadering praktisch niet mogelijk.

4. Data en methoden

Ik zal de hierboven beschreven constructie-specifieke evaluatie uitvoeren voor het zoeken naar vijfledige werkwoordsclusters, om te bepalen of dit enigszins succesvol te doen is. Hierbij gebruik ik de eerste drie benaderingen.

Om dit onderzoek uit te voeren, maak ik gebruik van het eerder genoemde PaQu (*Parse and Query*, Odijk et al. 2017) en van DACT (de Kok 2010). Het onderliggende zoeksysteem van deze twee methoden is dezelfde, en beide ondersteunen geavanceerde zoekmogelijkheden met XPath 2.0. Hiervan heb ik gebruik gemaakt om vijfledige werkwoordsclusters te vinden, door XPath 2.0-queries op te stellen die de gewenste structuren beschrijven.

Daarnaast kan PaQu via het web gebruikt worden en biedt het voor de minder technisch onderlegde onderzoeker een goed te doorgronden gebruikersinterface, waarbij ook queries opgesteld kunnen worden zonder kennis van XPath. Elke zin die aan een query voldoet, wordt door PaQu weergegeven in een lijst, waarna deze lijst gedownload kan worden voor verdere analyse. Ook is het mogelijk om tellingen uit te voeren, mochten er zo veel resultaten zijn dat handmatige analyse onpraktisch is. PaQu biedt een handleiding bij het opstellen van XPath-queries door middel van voorbeelden, en ook is het mogelijk om met behulp van een andere

corpuszoekmachine, GrETEL (Augustinus et al. 2013; Odijk, Van der Klis & Spoel 2018), queries te genereren door simpelweg een voorbeeldzin in te voeren waar de gewenste syntactische structuur in voorkomt. GrETEL heeft unieke voordelen qua gebruiksvriendelijkheid, en de nieuwste versie van GrETEL[5] bevat ook grote, automatisch geannoteerde corpora en de mogelijkheid om eigen corpora toe te voegen, maar niet de mogelijkheid om macro's (voorgedefinieerde stukjes query) te gebruiken. Deze versie was nog niet beschikbaar tijdens de uitvoering van dit onderzoek

Bij web-gebaseerde software als GrETEL en PaQu is het nadeel dat ingewikkelde zoekopdrachten op grotere corpora, die lang duren, potentieel vast kunnen lopen doordat alles via de webbrowser loopt. Dit kan zorgen voor problemen door verbroken verbindingen, automatisch uitgelogd raken, *timeouts* van verbindingen, onderbroken downloads, browser crashes en dergelijke. Ook is men er op deze manier van afhankelijk dat alle infrastructuur aan de kant van het beherende instituut goed werkt. Bij ingewikkelde zoekopdrachten, zoals naar vijfledige werkwoordsclusters, kan dit ervoor zorgen dat niet het gehele corpus doorzocht wordt.

DACT is moeilijker te gebruiken omdat het op de eigen computer geïnstalleerd moet worden, en omdat de onderzoeker daarvoor ook een corpus zoals Lassy Groot in eigen beheer moet hebben, maar het biedt wel directere toegang en is betrouwbaar genoeg om een groot corpus in zijn geheel te kunnen doorzoeken. Vanwege de betrouwbaarheid zal ik in dit artikel resultaten uit DACT rapporteren, maar PaQu biedt in theorie dezelfde mogelijkheden.

In PaQu zijn standaard twee grote, automatisch geannoteerde corpora beschikbaar, die beide onderdeel zijn van het eerder genoemde Lassy Groot-corpus en eveneens met DACT doorzocht kunnen worden: het NL-wiki corpus, van 8.7 miljoen zinnen, en het krantencorpus, bestaande uit bijna 15 miljoen zinnen. Het NL-wiki corpus is een kopie van de gehele Nederlandstalige Wikipedia zoals zij was op 4 augustus 2011, dat geannoteerd is met de Alpino-parser. Het krantencorpus bestaat uit het materiaal in Lassy Groot dat uit kranten komt: het kranten-deel van het SONAR500 corpus, en het Eindhoven-corpus. Van deze twee corpora heb ik gebruik gemaakt voor dit onderzoek. Helaas maken deze corpora geen onderscheid tussen Nederlands uit Vlaanderen en Nederlands uit Nederland, dus onderzoeksvragen die hierop betrekking hebben kunnen met deze meegeleverde corpora niet beantwoord worden. Ter vergelijking zullen we ook

5 GrETEL 4.1, http://gretel.ivdnt.org (Vandeghinste & Mertens 2020).

het handmatig geannoteerde Lassy Klein-corpus meenemen, waarvan geclaimd is dat het geen vijfledige werkwoordsclusters bevat.

5. Resultaten

Hieronder zal ik de besproken evaluatiemethoden zo goed als mogelijk toepassen op het zoeken naar vijfledige werkwoordsclusters met behulp van DACT (De Kok 2010) of PaQu (Odijk et al. 2017). Hiermee hoop ik een voorbeeld te geven voor andere onderzoekers die corpusdata van specifieke constructies willen verzamelen – voor het gebruik van PaQu is weinig technische kennis vereist, afhankelijk van hoe ingewikkeld de onderzochte constructie is. Vijfledige werkwoordsclusters zijn behoorlijk ingewikkeld en zijn niet via de basis-zoekinterface te vinden, omdat daar slechts naar een relatie tussen twee woorden gezocht kan worden. Wegens de eerder genoemde beperkingen van de toegang tot PaQu via het web was het niet mogelijk om deze zoekopdrachten op de volledige corpora uit te voeren, omdat deze zoekopdrachten zo complex zijn dat de server meerdere dagen nodig heeft om ze uit te voeren. Ik rapporteer daarom hieronder de resultaten van het zoeken met DACT, tenzij anders aangegeven.

5.1 Handmatige evaluatie van de zoekresultaten

Bij deze benadering worden de teruggegeven resultaten van een syntactische zoekopdracht handmatig beoordeeld. Elk resultaat dat overeenkomt met de zoekopdracht maar naar mijn oordeel geen geval van de doelconstructie is, wordt als een fout gemarkeerd. Hiermee kunnen we de precisie van de annotatie (zoals weergegeven in de zoekopdracht) meten. De constructies worden met syntactisch zoeken opgevraagd, waarbij de onderzoeker de verwachte syntactische structuur invoert als zoekopdracht zoals geoperationaliseerd binnen het syntactisch formalisme waar het corpus mee geannoteerd is. Dit is de gebruikelijke manier om in een treebank te zoeken.

Om werkwoordsclusters te vinden, moeten we ze definiëren, want er is geen werkwoordscluster-label in de annotatie van de Lassy-corpora (Van Noord, Schuurman & Bouma 2010). Wat we wel kunnen zien, is of een werkwoord een ander werkwoord als hoofd of complement heeft. Een definitie die hierop gebaseerd is, kan echter niet altijd omgaan met de mogelijkheid van clusterdoorbreking. Mijn zoekopdracht omvat daarom een vrij brede definitie van werkwoordsclusters: het moet gaan om een reeks van 5 werkwoorden in een bijzin die onderdeel zijn van een verbaal complement

(of een paar andere mogelijke constituenttypen zoals conjuncten), waaronder één finiet werkwoord, en waarbij alle bij dezelfde bijzin horen. De noties bijzin/hoofdzin zijn niet als zodanig gedefinieerd in de annotatie, dus dit is gespecificeerd als een reeks van constituent-typen, waaronder werkwoordsfinale bijzinnen (SSUB), infinitiefgroepen met *om te*, en *van*-complementen. Deze definities, en de technische operationalisatie ervan, zijn gebaseerd op die in eerder werk (Bloem, Versloot & Weerman 2014). De resultaten hiervan zijn voor het NL-wiki corpus en het krantencorpus te zien in Tabel 2. Bij de zoekopdracht naar de 1-2-3-4-5-volgorde specificeren we dat het finiete werkwoord van het rijtje in de lineaire volgorde op de eerste positie moet staan, bij de zoekopdracht naar de 5-1-2-3-4-volgorde moet deze op de tweede positie staan. Een zoekopdracht waarbij het finiete werkwoord op de laatste positie moet staan (om de 5-4-3-2-1-volgorde te vinden) leverde niets op.

Tabel 2 **Aantal gevonden kandidaat-werkwoordsclusters van 5 werkwoorden, en het aantal fouten**

Volgorde	NL-wiki		Kranten		Totaal		% accuratesse
	goed	fout	goed	fout	goed	fout	
1-2-3-4-5	9	6	6	3	15	9	37.5%
5-1-2-3-4	5	19	0	19	5	38	20.9%

In de kolommen 'goed' staat het aantal resultaten vermeld dat volgens mij een vijfledig werkwoordscluster is, en in de kolom 'fout' het aantal resultaten dat dat niet is. We zien dat er wel degelijk vijfledige werkwoordsclusters te vinden zijn, maar ook dat er veel valse positieven teruggegeven zijn: een precisie van 20.9% voor de 5-1-2-3-4-volgorde, en 37.5% voor de 1-2-3-4-5-volgorde. Een voorbeeld van een echt vijfledig werkwoordscluster is (6), in de 1-2-3-4-5-volgorde. (7) is een van de fouten:

(6) Dat bracht hem, niet beseffend dat de industrieën naar de Oeral geevacueerd zouden worden, tot het bevel in september eerst de saillant van Kiev af te snijden, terwijl een onmiddellijke opmars naar het oosten Moskou nog voor de herfstregens **zou hebben doen laten vallen**. [wik_part0070/43320-50-2][6]

6 De markeringen tussen vierkante haken verwijzen naar de unieke zinsnummers in het Lassy Groot-corpus, en de lezer kan die gebruiken om een specifieke voorbeeldzin terug te vinden in het corpus. Deze zin met het ID wik_part0070/43320-50-2 kan teruggevonden

(7) Hij kwam er achter dat alle verwijzingen die Ra'ul **zouden kunnen beschuldigen geschrapt waren.** [wik_part0171/193286-4-6]

Deze beide voorbeelden zijn teruggegeven bij de zoekopdracht naar 1-2-3-4-5-volgorde, met het finiete werkwoord als eerste in de lineaire volgorde. Het goede voorbeeld, voorbeeld (6), is voor mij van twijfelachtige grammaticaliteit, maar is correct geannoteerd: *zou* is het syntactisch hoofd van de bijzin, met een verbaal complement dat *hebben* als hoofd heeft, wat weer een verbaal complement heeft dat *doen* als syntactisch hoofd heeft, enzovoort, met het gehele cluster in één geschakelde reeks afhankelijkheden. Dit is de verwachte annotatie van een enkel groot werkwoordscluster in het annotatieformaat dat door de Alpino-parser wordt gebruikt. In het foute voorbeeld zien we een bijzin die ingebed is in een andere bijzin, waardoor alle werkwoorden op het einde bij elkaar staan. Dit voldoet aan mijn zoekopdracht, want ik zocht naar groepen werkwoorden in dezelfde bijzin, en alle werkwoorden in deze zin vallen inderdaad onder de hoogste bijzin, *dat alle verwijzingen*.... Hier had mijn definitie van werkwoordsclusters en dus de zoekopdracht specifieker moeten zijn: de werkwoorden moeten niet alleen in dezelfde bijzin staan, maar de syntactisch laagste bijzin waar ze onder vallen, moet dezelfde zijn voor alle werkwoorden.

Het gaat hier dus om een zoekfout, niet om een annotatiefout: de zoekopdracht omlijnde niet goed wat een vijfledig werkwoordscluster is, terwijl de annotatie deze reeks correct weergeeft als een drieledig en een tweeledig werkwoordscluster.

Tabel 3 Aantal gevonden kandidaat-werkwoordsclusters van 5 werkwoorden, en het aantal fouten

Volgorde	NL-wiki		Kranten		Totaal		% accuratesse
	goed	fout	Goed	fout	goed	fout	
1-2-3-4-5	9	0	6	0	15	0	100%
5-1-2-3-4	5	19	0	0	5	1	83.3%

Om dit op te lossen moet een bepaling in de zoekopdracht toegevoegd worden om te specificeren dat voor alle werkwoorden de syntactisch laagste bijzin dezelfde moet zijn. Dit heb ik gedaan, en de nieuwe resultaten staan

worden op basis van zijn zinsnummer met behulp van de PaQu-zoekopdracht die uitgevoerd wordt wanneer deze link geopend wordt:
https://paqu.let.rug.nl:8068/xpath?db=lassywiki&xpath=%2Falpino_ds%2Fsentence%5B%40sentid%3D%2243320-50-2%22%5D&mt=std&xn=20

in Tabel 3.[7] Hiermee verdwijnen bijna alle foute gevallen: er is nog één incorrect resultaat over bij de 5-1-2-3-4-zoekopdracht. Dat is het volgende geval:

(8) In de eerste plaats zijn er als AN aanvaarde, deels regionale constructies als 'ik ben wezen fietsen' en de befaamde zin 'ik zou jou wel eens **hebben willen zien durven blijven staan kijken'**, die overigens ook voor een aanzienlijk deel van de Nederlanders niet grammaticaal is, maar binnen het AN zeker mogelijk. [wik_part0445/864271-62-2]

Hier gooien de taalkundigen roet in het eten met een mogelijk geconstrueerde, metalinguïstische zin. Het gaat hier om een zevenledig werkwoordscluster in een hoofdzin, maar omdat de hoofdzin binnen een bijzin wordt geciteerd, voldoet dit aan de zoekopdracht waarin we in bijzinnen zoeken. Overigens is dit geval door Alpino wel correct geannoteerd op de wijze van een werkwoordscluster, namelijk als een aaneenschakeling van verbaal complementen. Door dit kunstmatige voorbeeld heeft deze zoekopdracht in 20 uit 21 gevallen het gewenste resultaat geleverd, met dus een precisie van 95.2%. We kunnen dus wel concluderen dat er over de precisie van de Alpino-parser op vijfledige werkwoordsclusters weinig te klagen is – met deze zoekopdracht naar vijf werkwoorden binnen dezelfde laagste bijzin

Tabel 4 Gevonden reeksen van hulpwerkwoorden

Werkwoorden	Aantal
zou moeten kunnen worden	8
moeten kunnen blijven worden	1
zou hebben doen laten	1
zou hebben kunnen worden	1
zou hebben moeten kunnen	1
zou hebben moeten zijn	1
zou kunnen gaan worden	1
zou moeten gaan worden	1
zou moeten hebben kunnen	1
zou moeten moeten worden	1
zou mogen gaan worden	1
zouden hebben willen zien	1
zouden moeten kunnen worden	1

7 De exacte queries voor mijn definities van werkwoordsclusters (na correctie), evenals de opgeleverde zoekresultaten, zijn te vinden in de Github-repository die bij dit paper hoort: https://github.com/bloemj/5verbclusters

worden, met één uitzondering, alleen vijfledige werkwoordsclusters teruggegeven. Tabel 4 geeft weer welke hulpwerkwoorden er gebruikt werden in de 20 correct gevonden vijfledige clusters. Het geval *zou moeten moeten worden* is waarschijnlijk een redigeerfout in de oorspronkelijke corpustekst.

In het Lassy Klein-corpus werden geen resultaten gevonden, wat overeenkomt met de bevindingen van Augustinus (2015).

5.1.1 Resultaten met PaQu

Zoals gezegd zijn deze zoekopdrachten zo complex dat het meerdere dagen kost om ze uit te voeren, wat problemen oplevert bij het gebruik van een webapplicatie als PaQu. Lang durende zoekopdrachten kunnen in de huidige versie van PaQu om technische redenen of door verbindingsproblemen onderbroken worden, met als gevolg dat niet het hele corpus doorzocht wordt. Tabel 5 laat zien dat PaQu hierdoor minder resultaten oplevert dan DACT voor dezelfde zoekopdracht, die wel het gehele corpus doorzoekt (maar daar ook dagen over doet). Het gaat hier om de eerstgenoemde zoekopdracht in deze paragraaf, die geen rekening houdt met bijzinnen die in elkaar ingebed zijn. Het gerapporteerde aantal resultaten voor PaQu is telkens het hoogste van drie zoekpogingen. Het aantal varieert per poging, omdat het afhangt van allerlei factoren wanneer de zoekopdracht onderbroken wordt. Dit technische probleem zal alleen optreden bij zeer complexe zoekopdrachten die op een groot corpus uitgevoerd worden, en zal het meeste onderzoek niet in de weg zitten, maar bij gebruik van PaQu is het wel aanbevolen om te controleren of de zoekopdracht wel volledig uitgevoerd wordt. Door zoekresultaten te evalueren weten we alleen nog niet wat we niet gezien hebben; we weten niet wat de *recall* is. De volgende twee benaderingen kunnen daar een antwoord op geven.

Tabel 5 Totaal aantal gevonden resultaten met PaQu en DACT met dezelfde zoekopdracht.

Volgorde	NL-wiki		Kranten		Totaal	
	DACT	PaQu	DACT	PaQu	DACT	PaQu
1-2-3-4-5	15	9	9	5	**24**	14
5-1-2-3-4	24	10	19	5	**43**	15

5.2 Terugvallen op eenvoudigere annotatielaag

Bij deze benadering negeren we de syntactische annotatie van het corpus en zoeken alleen op eenvoudigere eigenschappen, met het doel gevallen die syntactisch verkeerd geannoteerd zijn toch te kunnen vinden. In een treebank zijn we normaal gesproken geïnteresseerd in de syntactische

afhankelijkheden, maar doorgaans zijn deze gebaseerd op lexicale informatie, zoals woordsoort, die ook terug te vinden is in de annotatie. We proberen dus om de doelconstructie te definiëren op basis van enkel lexicale eigenschappen. Als we werkwoordsclusters willen vinden zonder gebruik te maken van syntactische structuren, is de meest logische optie om naar een reeks werkwoorden te zoeken. Op deze manier kunnen werkwoordsclusters gevonden worden die niet de juiste syntactische structuur toegewezen hebben gekregen.

Tabel 6 **Aantal gevonden reeksen van 5 werkwoorden, en het aantal daarvan dat daadwerkelijk een werkwoordscluster is.**

Volgorde	NL-wiki		Kranten		Totaal		% accuratesse
	goed	fout	goed	fout	goed	fout	
V-V-V-V-V	15	374	6	198	21	572	3.54%

In Tabel 6 staan de resultaten die niet door de syntactische zoekopdracht gevonden zijn, maar wel door deze zoekopdracht gebaseerd op woordsoort: 593 reeksen van vijf werkwoorden in totaal. Ook in het (gehele) Lassy Klein-corpus waren vier reeksen van vijf werkwoorden te vinden, maar geen hiervan was een vijfledig werkwoordscluster. In de meeste gevallen ging het om een zin zoals die in voorbeeld (7) hierboven, waarin meerdere kleinere werkwoordsclusters naast elkaar staan en daarmee een reeks van vijf werkwoorden vormen, maar geen cluster. Zonder gebruik te maken van de syntactische informatie is het niet mogelijk om deze gevallen van echte vijfledige clusters te onderscheiden. In de automatisch geannoteerde data vinden we ook vreemdere gevallen, zoals voorbeeld (9):

(9) Vergrendelen - Bewerken - Ontgrendelen. [wik_part0008/1860-10-1]

Hier lijkt het te gaan om een deel van de Wikipedia-gebruikersinterface dat in het corpus terecht is gekomen. Omdat het gaat om een reeks werkwoorden, wordt ook dit gevonden. Bijzonder is dat de streepjes ook als werkwoord aangemerkt zijn in de annotatie.

Na handmatige controle blijkt dat 21 van de 593 resultaten een vijfledig werkwoordscluster bevat. 20 hiervan zijn dezelfde gevallen die ook met syntactisch zoeken gevonden werden, maar toch was het niet helemaal voor niets om deze 593 zinnen handmatig te bekijken, want één van de correcte gevallen is nieuw:

(10) Daarbij zijn er in verschillende tradities ontstaan en in de periode van de papieren heraldiek schreven de herauten en wapenkoningen voor dat de rang van de drager ook aan de helm **zou moeten worden kunnen herkend**. [wik_part0089_64326-6-3]

Ten eerste is dit een 1-2-4-3-5-volgorde, maar dat had het vinden hiervan niet hoeven te verhinderen, want we hadden bij de syntactische zoekopdracht alleen maar gespecificeerd dat het syntactisch hoofd vooraan moet staan, zonder de onderlinge volgorde van de andere groepsvormende werkwoorden te specificeren. Het probleem is de ongrammaticaliteit van het begin van de zin. Dit is een ongrammaticale zin waarbij het automatisch annoteren volledig mislukt lijkt te zijn – de parser heeft deze zin waarschijnlijk niet volledig kunnen ontleden volgens de eigen grammatica, waardoor niet de gehele zin een structuur toegewezen heeft gekregen. Het deel dat wel herkend is, is geannoteerd als een discourse unit, een element dat geen syntactisch verband houdt met de rest van een zin (dit is normaal iets uit gesproken taal). Aan deze discourse unit is *kunnen herkend* toegewezen, en *zou moeten worden* is niet aan de structuur toegewezen.[8] Dit werkwoordscluster had niet met behulp van een syntactische zoekopdracht gevonden kunnen worden, omdat het niet de juiste syntactische structuur heeft gekregen in het corpus en daarmee niet zal voldoen aan een zoekopdracht naar syntactisch van elkaar afhankelijke werkwoorden.

Hiermee blijkt dat door annotatiefouten de *recall* van de zoekopdracht naar vijfledige werkwoordsclusters uit de vorige paragraaf niet perfect is: bij de 20 vijfledige werkwoordsclusters uit de syntactische zoekopdracht moeten we dus nog eentje optellen. Van die 21 waren er dus 20 correct geannoteerd in het corpus en gevonden met syntactisch zoeken, een recall van 95.2%. Een volledig overzicht van alle 21 gevonden gevallen is te vinden in Appendix 1. Nog steeds kan het echter zijn dat er gevallen niet gevonden zijn, bijvoorbeeld als de lexicale annotatie ook niet klopt. De laatste benadering omzeilt de annotatie volledig, en neemt alleen maar aan dat de woorden correct gespeld, getypt of getranscribeerd zijn.

5.3 Naar specifieke gevallen zoeken

Bij deze benadering zoeken we naar een woordenreeks die een specifiek geval is van de doelconstructie. Als we naar specifieke gevallen van de

8 Een afbeelding van de toegewezen syntactische structuur van deze zin is via DACT te zien: https: //paqu.let.rug.nl:8068/tree?db=lassywiki&names=true&mwu=false&arch=/net/corpora/LassyLarge/NLWIKI20110804/DACT/wik_part0089.dact&file=64326-6-3.xml.

constructie zoeken, zijn we niet afhankelijk van de annotatie. De onderzoeker kan een representatief geval van de constructie nemen, of een deel daarvan, en naar die combinatie van tekens zoeken in het corpus, net als wanneer men in een tekstbestand zoekt. In DACT en PaQu kan dit gedaan worden door op de *word*-eigenschap in de lexicale annotatie te zoeken. Deze eigenschap zal letterlijk overeen komen met de betreffende tekst, zelfs wanneer het om een niet-bestaand of fout gespeld woord gaat. Bij dit onderzoek naar vijfledige werkwoordsclusters zou een logische keuze de letterreeks *zou moeten kunnen worden* zijn. Dit was in Tabel 4 de meest frequente reeks van hulpwerkwoorden. We kunnen het ook nog iets minder specifiek maken door naar een reeks van de lemma's *zullen, moeten, kunnen, worden* te zoeken, om te generaliseren over verbuigingen van het finiete werkwoord. Het heeft weinig zin om er nog een hoofdwerkwoord aan toe te voegen – met een zeldzame constructie als deze is het zeer onwaarschijnlijk dat een specifiek vijfledig werkwoordscluster meer dan één keer te vinden is.

Met dit letterlijke type zoekopdracht zullen we natuurlijk slechts een beperkt deel van alle mogelijke vijfledige werkwoordsclusters kunnen vinden, maar in een groot corpus zullen dit er alsnog een redelijk aantal zijn. Net als in de vorige paragraaf kunnen we de resultaten hiervan vergelijken met die van de syntactische zoekopdrachten om te zien of we daarmee gevallen gemist hebben. Een ander voordeel van deze benadering is dat een *string search* minder complex is en sneller uitgevoerd kan worden. Dit betekent dat het met PaQu mogelijk is om een groter deel van het corpus te doorzoeken.

Een nadeel van deze benadering is dat we maar moeten aannemen dat de resultaten voor *zullen moeten kunnen worden* representatief zijn voor de rest van de vijfledige werkwoordsclusters. Als *zullen moeten kunnen worden* wel goed gaat, maar *zullen hebben doen laten* niet, dan zullen we dat op deze manier niet vinden, behalve door er expliciet naar te zoeken. We kunnen misschien wel een paar andere werkwoordscombinaties uitproberen, maar niet alle mogelijke.

Tabel 7 laat de resultaten hiervan zien. Er zijn 9 gevallen, waarvan 8 met *zou* en 1 met *zouden*. Dit komt overeen met de resultaten van de syntactische zoekopdracht, in Tabel 4. De syntactische zoekopdracht heeft dus geen gevallen gemist die we op deze manier wel hadden kunnen vinden, en we kunnen stellen dat de *recall* voor vijfledige werkwoordsclusters van het type *zullen moeten kunnen worden* 100% is.

Tabel 7 **Aantal gevonden reeksen van de lemma's *zullen moeten kunnen worden***

Volgorde	NL-wiki		Kranten		Totaal		% accuratesse
	goed	fout	Goed	fout	goed	fout	
zullen-moeten-kunnen-worden	6	0	3	0	9	0	100%

6. Discussie

Het blijkt wel degelijk mogelijk om natuurlijke taaldata voor vijfledige werkwoordsclusters te vinden, namelijk in grote automatisch geannoteerde tekstcorpora. Ondanks dat het gaat om een complexe constructie, worden vijfledige werkwoordsclusters behoorlijk goed geannoteerd. De Alpino-parser wijst niet snel een vijfledige werkwoordsclusterstructuur toe aan iets dat dat niet is. Het komt wel voor dat een vijfledig werkwoordscluster op een andere manier geannoteerd wordt, maar niet vaak – dit was bij 1 van de 21 gevonden gevallen zo.

Wel zijn de analysemogelijkheden met 21 resultaten, waaronder 15 in dezelfde volgorde, beperkt. We kunnen zeggen dat de 5-1-2-3-4-volgorde (tangconstructie) het meest frequent is, met 15 van de 21 gevallen. De 1-2-3-5-4-volgorde, die op basis van Stroop (2009) in Vlaanderen de voorkeur zou kunnen hebben, is niet aangetroffen. Verder is *zou moeten kunnen worden* verreweg de meest frequente combinatie van hulpwerkwoorden.

De bijzinslengte (exclusief het werkwoordscluster) is bij deze zinnen met vijfledige clusters gemiddeld 5.86 woorden, terwijl dit bij zinnen met tweeledige clusters 6.1 woorden is (Bloem, Versloot, en Weerman 2017), geen spectaculair verschil. De bijzinnen van de 5-1-2-3-4-volgorde zijn gemiddeld iets langer met 6.1 woorden, terwijl het gemiddelde 5.2 woorden is voor de 1-23-4-5-volgorde. Bij zinnen met een prepositioneel object die geen onderdeel is van het predicaat, en dus door extrapositie achter het werkwoordscluster geplaatst kunnen worden, vindt deze extrapositie plaats in 4 van de 5 gevallen. Volgens Willems & De Sutter (2015) is de positie na het werkwoordscluster de meest frequente standaardvolgorde, en dat geldt dus blijkbaar ook in het geval van vijfledige werkwoordsclusters. Opvallend is wel dat alle 4 gevallen met extrapositie in de 5-1-2-3-4-volgorde staan, terwijl het geval zonder extrapositie in de 1-2-3-4-5-volgorde staat, maar dit zijn te weinig gegevens om conclusies aan te kunnen verbinden.

Wat daarnaast nog opvalt, is dat er vrij veel complexe werkwoorden als hoofdwerkwoorden gebruikt zijn (*overwinnen, doorbreken*), waaronder ook vijf scheidbare werkwoorden (*tegenhouden, invoegen, kapotschieten,*

opnemen, voortbewegen), in beide volgorden. Die vijf werkwoorden tekenen voor 24% van alle gevallen. Twee ervan staan in de 1-2-3-4-5-volgorde, waar maar zes voorbeelden van waren, dus 33% van de gevallen. Bij tweeledige werkwoordsclusters zijn slechts 2% van de hoofdwerkwoorden scheidbaar, en is gevonden dat scheidbare werkwoorden relatief vaker in de 1-2-volgorde gebruikt worden, en dus aan het einde van het cluster geplaatst worden (De Sutter 2005). Het zal overbodig zijn om te zeggen dat deze verschillen niet statistisch significant zijn met deze kleine aantallen. Met enige creativiteit kunnen we hiermee een hypothese opstellen dat het hoofdwerkwoord eerder achteraan zal komen als het scheidbaar of op andere wijze complex is, of in complexe contexten, maar er is nog te weinig data om hier veel over te zeggen. Dit zou wel aansluiten bij mijn eerdere stelling over tweeledige werkwoordsclusters, namelijk dat informatie beter over de zin gespreid wordt wanneer het hoofdwerkwoord achteraan staat, wat de verwerking vergemakkelijkt (Bloem 2016b). Inhoudelijk gezien kunnen we over dit onderzoek weinig anders concluderen dan dat vijfledige werkwoordsclusters wel degelijk in natuurlijke taal bestaan, dat er minstens twee mogelijke woordvolgorden zijn en dat voor een kwantitatieve analyse van vijfledige werkwoordsclusters de corpora nog iets groter moeten worden. Het zou dan interessant zijn om te kijken waarom er volgordevariatie optreedt en of door die variatie informatie inderdaad beter door de zin gespreid wordt.

De drie evaluatiemethoden die ik voor dit onderzoek toegepast heb, mogen dan wel enig handmatig controlewerk vergen, ze bieden wel meer duidelijkheid over de kwaliteit van de annotatie voor het doel van een specifiek taalkundig onderzoek, zoals dit onderzoek naar vijfledige werkwoordsclusters. Ik kon hiermee zowel de *precision* bepalen als de *recall* inschatten bij het zoeken naar vijfledige werkwoordsclusters, en daarmee hopelijk de zorgen wegnemen die mensen vaak hebben bij het gebruik van automatische annotatie, vooral voor het onderzoeken van zo'n complexe constructie. Aangezien de drie benaderingen verschillende voordelen en nadelen hebben, lijkt het nuttig om ze tegelijk te gebruiken. Met de handmatige evaluatie van de resultaten kan de precisie van de zoekopdracht bepaald worden, en met het terugvallen op eenvoudiger annotatie kan de recall ingeschat worden, Bij het zoeken naar specifieke gevallen kan dit ook op een meer accurate, maar ook meer beperkte manier, al vond ik met deze derde benadering bij dit onderzoek geen extra gevallen.

Overigens zijn deze drie benaderingen niet slechts toepasbaar op taalkundig onderzoek. Ook in ander digitaal geesteswetenschappelijk onderzoek is zoeken in geannoteerde datasets vaak een belangrijk onderdeel van

de methodologie, en kunnen de hier beschreven benaderingen mogelijk ook toegepast worden. Bijvoorbeeld in de filosofie, waar semantisch zoeken toegepast wordt als hulpmiddel bij het uitvoeren van conceptuele analyses (Van Wierst et al. 2018), en geëvalueerd kan worden door met *string search* naar specifieke gevallen te zoeken.

Voor het Nederlands zijn een aantal automatisch geannoteerde corpora relatief eenvoudig beschikbaar via het web met behulp van PaQu, Momenteel zijn dit de Wikipedia-artikelen uit Lassy Groot, de krantenteksten uit Lassy Groot, de Nederlandstalige Childes-corpora, het Wablieft-corpus van eenvoudig te lezen Nederlands en het Eindhoven-corpus. Iedereen kan deze lijst uitbreiden, want in PaQu kan ook eigen tekst automatisch geannoteerd worden. Wel bleek het lastig om met deze web-interface complexe zoekopdrachten op een groot corpus uit te voeren, omdat deze lang kunnen duren. Voor een grondig onderzoek naar een complexe constructie is het aan te bevelen om lokaal geïnstalleerde software zoals DACT te gebruiken, of de daarbij horende API[9] of Python bindings[10], aangezien dit zonder problemen wekenlang aan kan staan. Hiermee kunnen mogelijk nog meer voorbeelden van vijfledige werkwoordsclusters in de hier gebruikte corpora gevonden worden.

Het gaat hier vooralsnog om corpora van geschreven taal. Er is geen automatisch verwerkt, groter equivalent van het Corpus Gesproken Nederlands, en daarmee geen gesproken corpus van de gewenste omvang voor dit onderzoek naar vijfledige werkwoordsclusters. Gesproken taal kan wel automatisch syntactisch geannoteerd worden, maar moet hiervoor eerst getranscribeerd worden. Hierdoor zal het niet om grote hoeveelheden tekst gaan, tenzij ook automatische transcriptie wordt toegepast. Het is mij niet bekend hoe ver de taaltechnologie voor het Nederlands met automatische transcriptie is, en er zijn naar mijn weten geen corpora beschikbaar die op deze manier verkregen zijn. Het stapelen van automatische methoden resulteert vaak ook in het stapelen van annoteerfouten, dus bij onderzoek met dit soort data zou constructie-specifieke evaluatie essentieel zijn.

Het is aan te bevelen om bij toekomstig taalkundig onderzoek met behulp van automatisch geannoteerde corpora ook de *precision* en *recall* te rapporteren, zoals gemeten met de hier besproken methoden. Hiermee wordt voor de lezer duidelijker of de data (en de daaruit volgende analyse) betrouwbaar zijn, en wanneer dit in meer artikelen gerapporteerd wordt,

9 https://github.com/rug-compling/alpinocorpus
10 https://github.com/rug-compling/alpinocorpus-python

zal ook blijken hoe goed automatische annotatie werkt voor allerlei constructies in het Nederlands die voor taalkundigen interessant zijn. Met dit soort gegevens zal de waarde van het gebruik van automatische annotatie als empirische methodologie voor de taalkunde duidelijker worden.

Hebben we nu al een corpus waar alle constructies in gevonden zouden moeten kunnen worden? Nee, zowel grammaticaliteitsoordelen als corpusdata zijn nodig om een volledige theorie van grammaticale beperkingen te vormen, want corpusdata kunnen altijd fouten bevatten en incompleet zijn (Pullum 2017). Wel is er in de steeds groter wordende corpora steeds meer te vinden, tot de eerder niet vindbare vijfledige werkwoordsclusters aan toe. In toekomstig werk wordt het dan ook tijd voor een studie naar de gebruikspatronen en volgordevariatie van zesledige werkwoordsclusters, wanneer we een paar jaar verder zijn en er nog grotere corpora beschikbaar zijn.

Referenties

Augustinus, Liesbeth (2015). *Complement raising and cluster formation in Dutch. A treebank-supported investigation*. Doctoraal proefschrift KU Leuven.

Augustinus, Liesbeth, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde (2013). Example-based treebank querying with GrETEL – now also for spoken Dutch. In: *Proceedings of the 19th Nordic Conference of Computational Linguistics* (*NODALIDA 2013*), Linköping University Electronic Press, 423-428.

Augustinus, Liesbeth, Vincent Vandeghinste & Frank Van Eynde (2012). Example-based treebank querying. In: *Proceedings of the eighth international conference on Language Resources and Evaluation* (*LREC*), 3161-3167.

Bastiaanse, Roelien & Gosse Bouma (2007). Frequency and linguistic complexity in agrammatic speech production. *Brain and Language* 103(1), 18-28.

van der Beek, Leonoor, Gosse Bouma & Gertjan van Noord (2002). Een brede computationele grammatica voor het Nederlands. *Nederlandse Taalkunde* 7(4), 353-374.

Blasi, Damian, Ryan Cotterell, Lawrence Wolf-Sonkin, Sabine Stoll, Balthasar Bickel & Marco Baroni (2019). On the distribution of deep clausal embeddings: A large cross-linguistic study. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 3938-3943.

Bloem, Jelke (2016a). Evaluating automatically annotated treebanks for linguistic research. In: *Proceedings of the 4th Workshop on Challenges in the Management of Large Corpora* (*CMLC-4*). Portorož, Slovenia: Institut für Deutsche Sprache, 8-14.

Bloem, Jelke (2016b). Testing the processing hypothesis of word order variation using a probabilistic language model. In: *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity* (*CL4LC*), 174-185.

Bloem, Jelke, Arjen Versloot & Fred Weerman (2014). Applying automatically parsed corpora to the study of language variation. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin: Dublin City University and Association for Computational Linguistics, 1974-1984.

Bloem, Jelke, Arjen Versloot & Fred Weerman (2017). Verbal cluster order and processing complexity. In: Enoch Aboh (red.), *Complexity in human languages: A multifaceted approach*. Elsevier, 94-119.

Bloemhoff, Henk (1979). Heranalyse van een Stellingwerver oppervlaktestructuur. *Us Wurk*, 28(1-4), 31-38.

Bouma, Gosse (2017). Finding long-distance dependencies in the Lassy corpus. *Crossroads Semantics: Computation, experiment and grammar*, 39-56.

Bouma, Gosse & Jennifer Spenader (2008). The distribution of weak and strong object reflexives in Dutch. *LOT Occasional Series* 12, 103-114.

Clark, Kevin & Christopher D Manning (2016). Improving coreference resolution by learning entity-level distributed representations. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), 643-653

Coussé, Evie, Mona Arfs & Gert De Sutter (2008). Variabele werkwoordsvolgorde in de Nederlandse werkwoordelijke eindgroep: een taalgebruiksgebaseerd perspectief op de synchronie en diachronie van de zgn. rode en groene woordvolgorde. In: Gudrun Rawoens (red.), *Taal aan den lijve: het gebruik van corpora in taalkundig onderzoek en taalonderwijs*. Academia Press, 29-47.

De Schutter, Georges (2012). De werkwoordelijke eindgroep en nog steeds geen einde? *Verslagen & Mededelingen van de Koninklijke Academie voor Nederlandse Taal- en Letterkunde* 122(1), 1-38.

De Sutter, Gert (2005). *Rood, groen, corpus! Een taalgebruiksgebaseerde analyse van woordvolgordevariatie in tweeledige werkwoordelijke eindgroepen*. Doctoraal proefschrift KU Leuven.

Gibson, Edward & Evelina Fedorenko (2013). The need for quantitative methods in syntax and semantics research. *Language and Cognitive Processes* 28(1-2), 88-124.

Haeseryn, Walter, Kirsten Romijn, Guido Geerts, Jaap de Rooij & Maarten Cornelis van den Toorn (1997). *Algemene Nederlandse Spraakkunst*. Tweede editie. Groningen: Martinus Nijhoff.

Hinrichs, Erhard & Kathrin Beck (2013). Auxiliary fronting in German: A walk in the woods. In: *The Twelfth Workshop on Treebanks and Linguistic Theories* (*TLT12*), 61-72.

Kakkonen, Tuomo (2005). Dependency treebanks: methods, annotation schemes and tools. In: *Proceedings of NODALIDA 2005*, 94-104.

de Kok, Daniël (2010). *Dact* [*Decaffeinated Alpino Corpus Tool*]. <http://rug-compling.github.com/dact>

Lin, Yuri, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman & Slav Petrov (2012). Syntactic annotations for the Google Books Ngram corpus. In: *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics, 169-174.

Müller, Frank Henrik. 2004. Stylebook for the Tübingen partially parsed corpus of written German (TüPP-D/Z). In: *Sonderforschungsbereich 441, Seminar für Sprachwissenschaft, Universität Tübingen* 28.

Napoles, Courtney, Matthew Gormley & Benjamin Van Durme (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics, 95-100.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. (2016). Universal Dependencies v1: A multilingual treebank collection. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (*LREC'16*), 1659-1666.

van Noord, Gertjan & Gosse Bouma (2009). Parsed corpora for linguistics. In: *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?* Association for Computational Linguistics, 33-39.

van Noord, Gertjan, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer Linde, Ineke Schuurman, Erik Tjong Kim Sang & Vincent Vandeghinste (2013). Large scale syntactic

annotation of written Dutch: Lassy. In: Peter Spyns & Jan Odijk (red.), *Essential Speech and Language Technology for Dutch, Theory and Applications of Natural Language Processing*. Berlin: Springer, 147-164.

van Noord, Gertjan, Ineke Schuurman & Gosse Bouma (2010). *Lassy Syntactische Annotatie*. Revision 21780, November 13, 2019.

van Noord, Gertjan (2009). Huge parsed corpora in LASSY. In: *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)* 12. LOT, 115-126.

van Noord, Gertjan, Piet Mertens, Cédric Fairon, A. Dister & P. Watrin (2006). At Last Parsing Is Now Operational. In: *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*. Leuven University Press, 20-42.

Odijk, Jan, Gertjan van Noord, Peter Kleiweg & Erik Tjong Kim Sang (2017). The Parse and Query (PaQu) application. In: Jan Odijk & Arjan van Hessen (eds.), *CLARIN in the Low Countries*. Ubiquity Press, 281-297.

Odijk, Jan, Martijn van der Klis & Sheean Spoel (2018). Extensions to the GrETEL Treebank Query Application. In: *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*. Praag: Charles University, 46-55.

Odijk, Jan (2015). Linguistic research with PaQu. *Computational Linguistics in the Netherlands journal* 5, 3-14.

Pullum, Geoffrey K (2007). Ungrammaticality, rarity, and corpus use. *Corpus Linguistics and Linguistic Theory* 3(1), 33-47.

Pullum, Geoffrey K (2017). Theory, data, and the epistemology of syntax. *Grammatische Variation. Empirische Zugänge und theoretische Modellierung*, 283-298.

Schäfer, Roland & Felix Bildhauer (2012). Building large corpora from the web using a new efficient tool chain. In: *LREC*, 486-493.

Stroop, Jan (1970). Systeem in gesproken werkwoordsgroepen. *Taal en Tongval* 22, 128-147.

Stroop, Jan (2009). Twee- en meerledige werkwoordsgroepen in gesproken Nederlands. In: Egbert Beijk e.a. (red.), *Fons Verborum. Feestbundel voor prof. dr. Fons Moerdijk*. Leiden: Instituut voor Nederlandse Lexicologie, 459-469.

Vandeghinste, Vincent & Koen Mertens (2020). GrETEL @ INT: Querying Very Large Treebanks by Example. *Computational Linguistics in the Netherlands 30*.

van Wierst, Pauline, Arianna Betti, Steven Hofstede, Thom Castermans, Michel Westenberg, Yvette Oortwijn, Shenghui Wang & Rob Koopman (2018). BolVis: Visualization for Text-based Research in Philosophy. In: *3rd Workshop on Visualization for the Digital Humanities*. Berlin.

Willems, Annelore & Gert De Sutter (2015). Reassessing the effect of the complexity principle on PP Placement in Dutch. *Nederlandse Taalkunde* 20(3), 339-367.

Over de auteur

Jelke Bloem, Universiteit van Amsterdam
E-mail: j.bloem@uva.nl

Appendix: gevonden vijfledige werkwoordsclusters

5-1-2-3-4-volgorde

(1) Vier kan gezien worden als de vier steden waartussen gereden zou moeten gaan worden (Amsterdam, Rotterdam, Antwerpen en Brussel). [wik_part0049/25506-4-6]
(2) Er zijn simpel gezegd twee manieren om spraak te herkennen: via een vaste grammatica waarbij de ontwerper bepaald welke woorden op welk moment herkend kunnen worden en via de " groot vocabulaire " spraakherkenning waarbij in principe alles herkend zou moeten kunnen worden. [wik_part0107/88347-26-1]
(3) Het is moeilijk te begrijpen hoe met het toenmalige primitieve systeem van seinvlaggen de communicatieproblemen bij het coördineren van een slaglinie van twaalf kilometer lengte overwonnen zouden hebben kunnen worden, vooral als we bedenken dat ook de Britten 24 ingehuurde koopvaarders gebruikten. [wik_part0134/125720-25-5]
(4) Homoseksualiteit werd nu niet meer gezien als een misdaad, maar als een medische of psychische aandoening,die genezen zou moeten kunnen worden. [wik_part0139/13627425-7]
(5) Deze stelling werd aanzien als een laatste zware stelling waar de vijand gedurende langere tijd tegengehouden zou moeten kunnen worden. [wik_part0142/140549-23-2]
(6) Recent moleculair onderzoek (suggereert dat de familie "Rafflesiaceae" ontstaan is vanuit de wolfsmelkfamilie, en daar misschien bij ingevoegd zou kunnen gaan worden. [wik_part0144/142731-7-1]
(7) Een succesvol offensief tegen de Grebbelinie was niet essentieel voor het welslagen van dit plan - - hoe dan ook zou een aanval daar de Nederlanders afleiden en hun krachten binden – maar het Duitse opperbevel ging ervan uit dat het Nederlandse leger zo zwak was dat de Grebbelinie snel doorbroken zou moeten kunnen worden, vooral omdat men abusievelijk veronderstelde dat de Nederlandse hoofdweerstand bij het oostfront van de Vesting Holland geboden zou worden, zoals inderdaad oorspronkelijk in de bedoeling gelegen had. [wik_part0210/265787-4-9]
(8) Clandestien ging op papier het onderzoek door naar een verbeterde versie, de SARL 42, die voorzien zou moeten moeten worden van de ARL 3-toren met 75 mm kanon en optische afstandsmeter. [wik_part0569/1215663-71-6]

(9) Ook in de luchtmacht deed bepantsering soms haar intrede; met name tijdens de Tweede Wereldoorlog, toen het de piloten moest beschermen en bij de ontwikkeling van zwaar bepantserde vliegtuigen, die in theorie moeilijker kapot geschoten zouden moeten kunnen worden door vuur vanaf de grond. [wik_part0615/1367504-4-4]

(10) De afgelopen weken leverden twee rechercheurs van het onderzoeksteam harde kritiek op hun collega's en superieuren, omdat die koste wat kost André de Vries veroordeeld zouden hebben willen zien. [wr-p-p-g_part00017/WR-P-P-G-0000009009.p.4.s.1]

(11) Dit morele dilemma (namelijk of zo'n film niet overal vertoond zou moeten kunnen worden) valt nog net binnen de grens van Etty. [WR-P-P-G_part00002/WR-P-P-G0000002165.p.2.s.3]

(12) Het gaat niet meer om knap gemaakte korte films, om dappere of laffe, om korte korte films of nog veel kortere, het gaat erom dat ze gemaakt moeten kunnen blijven worden, en vertoond. [WR-P-P-G_part00136/WR-P-P-G-0000071554.p.2.s.2]

(13) 'Redelijk onkuis ', en ' een oneerlijke vraag ', vindt Josien Folbert de vraag van Janse aan Nederlandse imams of er in Mekka niet een kerk gebouwd zou moeten kunnen worden. [WR-P-P-G_part00230/WR-P-P-G-0000111585.p.15.s.1]

(14) Vandaar dat er al langer een debat bezig is over de vraag of die souvereiniteit soms niet gebroken zou moeten kunnen worden door ' humanitaire interventies '. [WR-P-PG_part00230/WR-P-P-G-0000111732.p.17.s.7]

(15) Bovendien wilden ze voorkomen dat voortaan gekwalificeerde havenarbeid (zoals laden en lossen) gedaan zou mogen gaan worden door ongekwalificeerde arbeidskrachten, met alle veiligheidsrisico's van dien. [WR-P-P-G_part00336/WR-P-P-G0000157997.p.5.s.6]

1-2-3-4-5-volgorde

(16) In sommige uitbeeldingen wordt het alter ego echter slechts schematisch voorgesteld, soms zelfs zodanig dat de uitbeelding van het alter ego er niet meer is, hoewel in de figuur het blok steen wel is behouden waar het alter ego zou hebben moeten zijn gebeeldhouwd. [wik_part0205/55088-29-9]

(17) Dat bracht hem, niet beseffend dat de industrieën naar de Oeral geevacueerd zouden worden, tot het bevel in september eerst de saillant van Kiev af te snijden, terwijl een onmiddellijke opmars naar het

oosten Moskou nog voor de herfstregens zou hebben doen laten vallen. [wik_part0070/43320-50-2]

(18) (Als voorbeeld werd hierbij gewezen op de hypothecaire lening die reeds geruime tijd zonder problemen wordt afbetaald, en derhalve tegen (ongeveer) de nominale waarde zou moeten kunnen worden opgenomen, doch die in een " distressed market " tegen een aanmerkelijk lagere koers zou moeten worden gewaardeerd) [wik_part0483/969839-563]

(19) Het dier was evident een tweevoeter, die zich met zijn enorme benen snel lopend zou moeten hebben kunnen voortbewegen, zij het dat de vlieghuid dit wellicht wat had belemmerd. [wik_part0687/1623942-21-2]

(20) Op Chili werden schelpen landinwaarts gevonden waaruit Darwin zou hebben moeten kunnen concluderen dat dit vroeger een zeebodem was en nu omhoog geduwd door snelle geologische processen (in plaats van door langzame veranderingen). [wik_part0731/1770058-11-5]

1-2-4-3-5-volgorde

(21) Daarbij zijn er in verschillende tradities ontstaan en in de periode van de papieren heraldiek schreven de herauten en wapenkoningen voor dat de rang van de drager ook aan de helm zou moeten worden kunnen herkend. [wik_part0089/64326-6-3]