TAALKENNIS IN TEKSTVERWERKING

Koenraad de Smedt, Carla Huls en Fieny Pijls
NICI
Universiteit van Nijmegen
Postbus 9104, 6500 HE Nijmegen

Samenvatting

Het gebruik van de computer voor de verwerking van natuurlijke taal heeft zich in het verleden sterk toegespitst op automatische vertaling en vraag-antwoord-systemen. Nochtans zijn er veel meer toepassingsgebieden waar natuurlijke taalverwerking interessante mogelijkheden biedt. De toepassing van taalkennis in redactionele taken is tot op heden onvoldoende gewaardeerd en geëxploiteerd. In dit artikel schetsen wij een auteursomgeving *die taalkundige ondersteuning biedt bij het schrijven en redigeren van teksten. Wij bespreken onder meer automatische correctie van tik- en spelfouten (ook grammatische zoals d/t-fouten), betrouwbare woordafbreking en raadpleging van een lexicon. Ook stellen we enkele afgeleide systemen voor, met name een schooltekstverwerker en een generator van semi-standaardteksten.*

1 Inleiding

Tekstverwerking is niet meer het exclusieve domein van typisten: steeds meer auteurs schrijven zelf hun wetenschappelijke artikelen, zakenbrieven, jaarverslagen of romans met een min of meer gespecialiseerde tekstverwerker. Toch moet een auteur voor de taalkundige aspecten van de tekst nog steeds papieren naslagwerken raadplegen zoals grammatica's en woordenboeken. Sterker zelfs, de mogelijkeden die een tekstverwerker biedt om willekeurige letterreeksen in te voegen, weg te halen of te verplaatsen leiden soms tot foute zinnen. Het ligt dan ook voor de hand om de machine meer te laten doen dan alleen maar een mooie rechtermarge produceren.

Kempen e.a. (1987) geven een visie op een nieuwe generatie van redactionele programmatuur waarin onder meer specifieke ondersteuning wordt geboden op gebied van de woordenschat, grammatica, spelling en stijl. Het doel van deze programmatuur is auteurs met één druk op de knop alle mogelijke hulpmiddelen ter beschikking te stellen waarmee zij hun gedachten optimaal in natuurlijke taal kunnen formuleren. Kempen e.a. noemen dit soort van interactieve programmatuur een *auteursomgeving*. Redactionele programmatuur die commentaar geeft op een tekst *nadat* die geschreven is, wordt meestal aangeduid met de term *text critiquing* (Richardson en Braden-Harder, 1988).

Wij beargumenteren dat de ontwikkeling van redactionele programmatuur met taalkennis een wetenschappelijk boeiende bezigheid is, omdat heel wat facetten van het menselijk taalvermogen

Verschijnt in: TTT, 8, nr. 3 (1988)

erbij aan bod komen. Daar echter niet voor elke taak al die facetten strikt noodzakelijk zijn, is het mogelijk om al snel tot toepasbare resultaten te komen. Tabel 1 verbindt bij wijze van voorbeeld verschillende niveaus van taalkennis met een aantal taken die met behulp daarvan uitgevoerd kunnen worden.

Tabel 1: Taken op verschillende niveaus van taalkennis.

1	*Lexicale* kennis:	detectie van tik- en spelfouten (niet in zinscontext); correctie van tikfouten; consistentie van spelling waarborgen; opzoeken van woorden in een electronisch woordenboek.
2	*Fonologische* kennis:	correctie van spelfouten (niet in zinscontext).
3	*Morfologische* kennis:	detectie en correctie van samenstellingen, afleidingen en inflectievormen; woordafbreking.
4	*Syntactische* kennis:	detectie en correctie van fouten tegen de grammatica (bijvoorbeeld tik- en spelfouten in de zinscontext); detectie van ambiguïteiten; grammaticaspreadsheet.
5	*Semantische* kennis:	oplossen van ambiguïteiten; stijlcorrectie; synoniemen.

In het volgende gaan wij eerst in op enkele van deze taken binnen een auteursomgeving die ontwikkeld wordt aan de K.U. Nijmegen in samenwerking met Océ-Nederland in het kader van ESPRIT Project OS-82. Daarna introduceren wij het basisontwerp van het systeem voorzover het de taalkundig interessante aspecten betreft. Verder geven wij details over specifieke modulen binnen dit ontwerp. Tenslotte schetsen wij enkele toekomstige toepassingen en uitbreidingen.

2 Taken binnen een auteursomgeving

2.1 Correctie

Eén van de meest tijdrovende taken bij de redactie van een tekst is die van de corrector. Deze moet niet alleen controleren op tik- en zetfouten (typografische fouten), maar ook op spelfouten (orthografische fouten) en op fouten tegen de grammatica (grammatische fouten[1], bijvoorbeeld de bekende d/t-fouten). Stijlfouten laten wij hier even buiten beschouwing.

Van Berkel en De Smedt (1988) brengen de verschillen tussen typografische en orthografische fouten in kaart. *Typografische* fouten zijn motorische fouten die voortkomen uit verkeerde toetsaanslagen. De kenmerken van deze fouten zijn afhankelijk van het gebruikte toetsenbord en niet van de gebruikte taal. Ongeveer tachtig procent van deze fouten zijn enkelvoudige fouten, waarbij de volgende soorten worden onderscheiden:

1 weglatingen: *totsenbord*
2 toevoegingen: *toeetsenbord*
3 substituties: *toitsenbord*
4 transposities: *teotsenbord*

De overige twintig procent zijn complexe fouten, dus combinaties van enkelvoudige fouten.

Orthografische fouten zijn cognitieve fouten die bestaan uit een substitutie van een foute spelling voor de correcte, waarbij de auteur de juiste spelling helemaal niet kent of vergeten is. Een belangrijk kenmerk van dit type fout is dat de uitspraak van de foute spelling meestal gelijk

is aan die van de goede, bijvoorbeeld *toetsenbort*. Deze fouten zijn dus afhankelijk van de correspondentie c.q. discrepantie tussen de spelling en de uitspraak binnen een taal.

De behandeling van orthografische fouten is volgens Van Berkel en De Smedt (1988) belangrijker dan die van typografische fouten. Spelfouten en grammaticafouten maken in het algemeen een slechtere indruk op de lezer, wellicht omdat zij wijzen op een gebrek aan kennis. Bovendien kunnen spelfouten, in tegenstelling tot tikfouten, in de regel niet door de auteur zelf ontdekt en gecorrigeerd worden. Terwijl voor tikfouten automatische detectie al voldoende is, is voor spelfouten automatische correctie een absolute noodzaak.

Een apart soort van fouten op woordniveau zijn de *grammatische* woordfouten. Hierbij gaat het om woordvormen die ongepast zijn binnen de zinscontext; het zijn meestal fouten tegen de congruentie[2]. *Grammatische revisiefouten* zijn een direct gevolg van het gebruik van een tekstverwerker bij de revisie van een tekst. Door het toevoegen, weghalen of veranderen van woorden in de zin ontstaan ongewild scheve zinnen. Beschouw de veranderingen van zin (1) in (1') en van zin (2) in (2').

(1) ...die voortkomen uit een verkeerde toetsaanslag.
(1')* ...die voortkomen uit een verkeerde toetsaanslag*en*.
(2) ...dat een tekstverwerker voor deze taak zeer geschikt is.
(2')* ...dat een tekstverwerkers voor deze taak zeer geschikt is.

Grammatische spelfouten zijn cognitieve fouten waarbij de zinscontext bepalend is voor het spellingsbeeld. D/t-fouten vallen in deze groep. Een voorbeeld van een veel voorkomende fout is het gebruik van *wordt* in zinnen als (3):

(3)* Wordt je al vier jaar?

Aangezien bij grammatische woordfouten de gebruikte woordvormen vaak wel bestaan, maar de zinscontext bepalend is voor het juiste gebruik van een vorm, is het voor detectie en correctie niet voldoende om woorden op te zoeken in een woordenlijst. Het is noodzakelijk om de zin geheel of ten dele te ontleden.

2.2 Grammaticaspreadsheet

In de wereld van persoonlijke computers is het *spreadsheet* (spreidblad) al een ingeburgerd begrip. Dit soort programma's wordt vooral gebruikt in de zakenwereld om het effect van een verandering van de waarde van een variabele in een vergelijking te zien. Neem bijvoorbeeld de vergelijkingen (4-6) die de relatie tussen een aantal variabelen in een boekhoudkundig systeem weergeven.

(4) netto winst = bruto winst - belasting
(5) bruto winst = omzet - kosten
(6) omzet = stukprijs * stuks

Met behulp van het spreadsheet is het mogelijk het effect te zien van bijvoorbeeld de verandering van de *stukprijs* op andere variabelen zoals *omzet, bruto winst* en *netto winst.* Een spreadsheet zal bij elke verandering namelijk trachten om het geheel consistent te houden volgens de regels van de vergelijkingen.

Het *grammaticaspreadsheet* (Kempen e.a., 1987) voert een vergelijkbare taak uit, maar dan op taalkundig niveau. Het kan veranderingen in een deel van een zin automatisch ‘doorberekenen’

naar andere delen om grammatische congruentie te waarborgen. Verandering van *boek* in zin (7) naar het meervoud maakt bijvoorbeeld de vier aanpassingen in (7') noodzakelijk.

(7) Het blauwe *boek* dat gisteren nog uitgeleend was, is terug.
(7') *De* blauwe boeken *die* gisteren nog uitgeleend *waren, zijn* terug.

Zelfs met behulp van zeer geavanceerde tekstverwerkers is dit een tijdrovende revisie, die gemakkelijk tot fouten leidt. Met behulp van het grammaticaspreadsheet wordt deze revisie op een eenvoudige manier snel en correct uitgevoerd. Het volstaat om één enkel woord te veranderen: de rest van de zin wordt automatisch aangepast.

2.3 Lexicon

De auteur kan vanuit de auteursomgeving een electronisch opgeslagen lexicon raadplegen. Doordat dit lexicon niet gebonden is aan de beperkingen van een gedrukt boek, kan het vele specifieke functies aanbieden, zoals het opzoeken van een willekeurige woordvorm (bijvoorbeeld *gelogen*) naast de gewone citatievorm van een woord (*liegen*). Dit is niet alleen nuttig voor auteurs die het Nederlands niet als moedertaal beheersen, het spaart ook tijd. Eenvoudig door het aanwijzen van een woordvorm die al op het scherm staat kan een auteur allerlei informatie laten verschijnen, zoals synoniemen of alle mogelijke inflectievormen, ook die welke niet in een gewoon woordenboek worden opgesomd. Uitbreiding van zo'n lexicon met morfologische kennis maakt het mogelijk inflectievormen te generen van niet bestaande, nieuwe of samengestelde woorden.

Door gebruik te maken van een systeem voor spelfoutcorrectie (zie verderop) kan de auteur woorden waarvan de spelling niet bekend is toch opzoeken, en wel door ze te schrijven "zoals je ze uitspreekt". Dit is zelfs mogelijk als de eerste letter niet bekend is, bijvoorbeeld *garisma* i.p.v. *charisma.* Het lexicon kan door een auteur interactief aangepast worden. Hij kan bijvoorbeeld lexica met vaktermen, eigennamen en plaatsnamen toevoegen.

Door de zinscontext te betrekken bij het opzoeken van een woord in de tekst kan het woord eventueel gedisambigueerd worden, waardoor de auteur slechts relevante informatie te zien krijgt. Als een auteur bijvoorbeeld een synoniem van *berichten* zoekt door dat woord aan te wijzen in zin (8), zal alleen een synoniem voor het zelfstandig naamwoord *berichten* gevonden worden en niet voor het werkwoord.

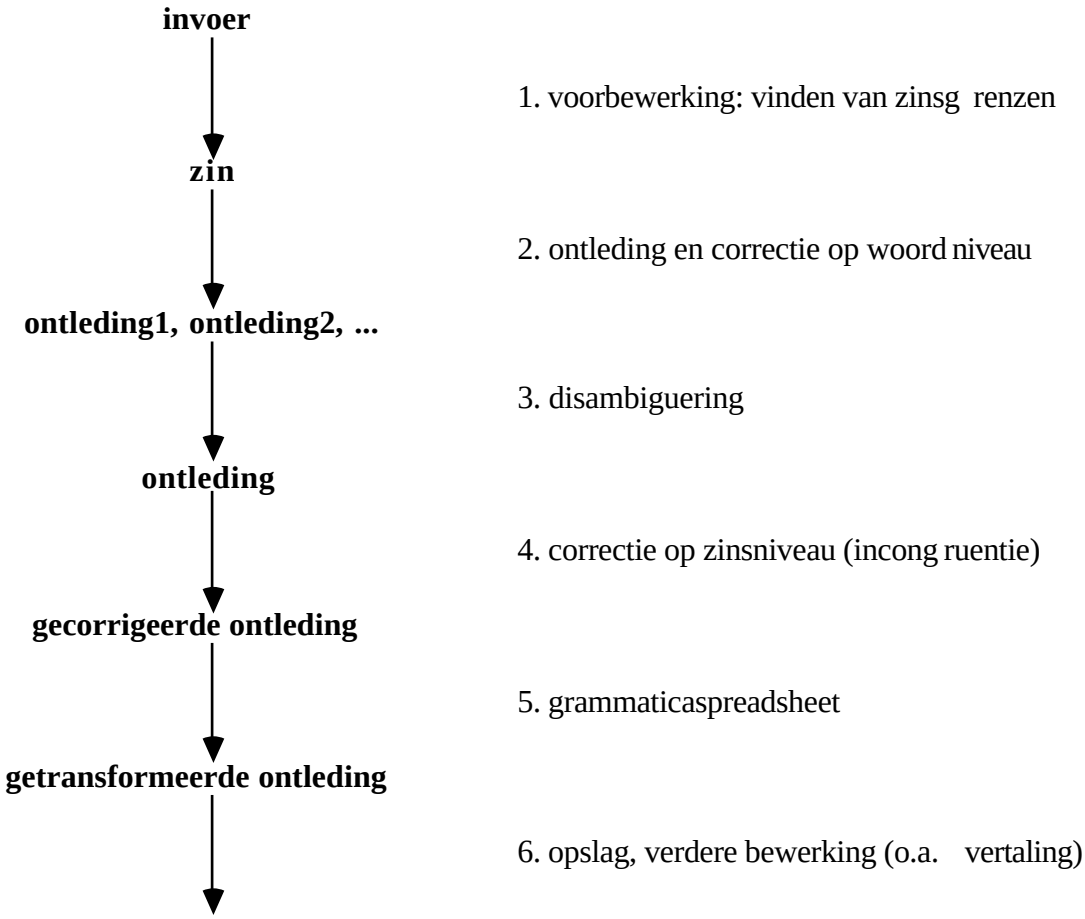
(8) De *berichten* uit het buitenland waren zeer gunstig.

Uiteraard vervult het lexicon zijn functie als informatieleverancier niet alleen voor de auteur, maar tevens voor de andere modulen in de auteursomgeving, bijvoorbeeld voor de ontleder en de lettergreepscheider. Van der Linden en De Smedt (1987) brengen de vereisten naar voren die aan zo'n lexicon worden gesteld.

3 Ontwerp van een auteursomgeving

3.1 Overzicht

Figuur 1 geeft in een schematisch overzicht de verschillende stadia van de verwerking van een zin weer.



Figuur 1: Overzicht van de informatiestroom in de taalkundige modulen.

Omdat voor de meeste taken (correctie, spreadsheet, enz.) kennis nodig is over de syntactische structuur van de betreffende zin, wordt deze eerst ontleed. In feite is er een wederzijdse afhankelijkheid tussen ontleding en correctie: om een zin te kunnen ontleden moeten alle woorden bekend (en dus gecorrigeerd) zijn, maar om de woorden te corrigeren moet zoveel mogelijk van de zinsstructuur bekend zijn. Deze circulariteit hebben wij enigszins kunnen oplossen door het gebruik van een bijzondere ontleder (zie verderop): ontleding en correctie op woordniveau geschieden in één enkel proces.

Het ontleedproces levert mogelijk verscheidene syntactische structuren op waartussen een keuze moet worden gemaakt. Bij syntactische ambiguïteit duiken vaak alternatieven op die semantisch onwaarschijnlijk zijn. Vergelijk bijvoorbeeld de structuren (9') en (9") voor zin (9).

(9) Zij nam haar hoed van de kapstok.
(9') Zij nam [haar hoed] [van de kapstok].
(9")? Zij nam [haar hoed van de kapstok].

Een semantisch interpretatiealgoritme zou de eerste structuur (9') kunnen prefereren. Wij hebben er voorlopig echter voor gekozen om de keuze tussen verschillende ontledingen geheel in handen van de gebruiker te laten. Hiertoe worden de zinnen grafisch, in de vorm van boomstructuren, aan de gebruiker gepresenteerd.Nadat de gebruiker een keuze heeft gemaakt door het aanwijzen van een structuur, kunnen incongruenties automatisch gecorrigeerd worden. In de gecorrigeerde zin kan de auteur met behulp van het grammaticaspreadsheet nog verdere wijzigingen

aanbrengen, waarna de tekst met de bijbehorende grammatische structuur klaar is voor opslag. Omdat de zin nu geheel ontleed en gedisambigueerd is, kan de opgeslagen zinsstructuur zonder verdere tussenkomst van de auteur worden bewerkt, bijvoorbeeld voor vertaling of parafrasering. In de volgende paragrafen zullen wij dieper ingaan op de verschillende taalkundige modulen in onze auteursomgeving.

3.2 Ontleding

Aangezien de ontleder foutieve zinnen moet kunnen ontleden, met name zinnen met tik- en spelfouten, met inbegrip van incongruenties, moeten ontleding en woordcorrectie op elkaar afgestemd zijn. Bovendien moet de ontleder deze fouten kunnen detecteren, diagnosticeren en zo mogelijk corrigeren. Hiertoe is in onze auteursomgeving een 'robuuste' ontleder geïmplementeerd, die gebaseerd is op werk van Konst (1986). Deze ontleder levert boomstructuren op die niet alleen de *immediate constituents* weergeven, maar ook de grammatische functies voor de woorden en woordgroepen identificeert.

De ontleder verwerkt de binnenkomende woorden van links naar rechts en *bottom up*. Bij ambiguïteiten worden alle mogelijke ontledingen pseudo-parallel verder verwerkt. De processen zijn zo gesynchroniseerd dat de ontleding strikt van links naar rechts en woord voor woord verloopt. Het *stapelgeheugen* wordt voor elke nieuwe keuze gedupliceerd, wat *backtracking* overbodig maakt. Elk nieuw ontleedproces dat zo ontstaat, krijgt een initiële grammaticaliteitswaarde van 1. Telkens wanneer een proces een syntactische fout detecteert, wordt de waarde van dit proces verlaagd met een van tevoren gedefinieerde factor. Een code voor de gevonden fout wordt dan in de stapel van de desbetreffende ontleding opgeslagen. Zodra de waarde van een proces beneden een vooraf bepaalde drempelwaarde komt, wordt het proces gestopt en de bijbehorende stapel verwijderd. Op deze manier werkt de drempelwaarde als een filter die het aantal ontledingen beperkt.

Wanneer de ontleder een onbekend woord vindt (mogelijk een tik- of spelfout), dan wordt uitgeprobeerd welke woordsoorten op die plaats in de zin passen. Door het filter blijven slechts enkele mogelijkheden over die dan gebruikt worden om het aantal kandidaten voor correctie in te perken. In zin (10) kan *wort* bijvoorbeeld alleen de persoonsvorm zijn: een mogelijke verbetering is dus *wordt* en niet bijvoorbeeld *word* of *worp.*

(10)* Hij wort 36 jaar.

De ontleder faalt niet op zinnen met congruentiefouten maar ontleedt ze volledig, zij het dat een code voor de gemaakte fout wordt opgeslagen in het stapelgeheugen van de ontleder. Zin (11) zal een ontleding opleveren met een code voor incongruentie van onderwerp en persoonsvorm.

(11)* Hij word 36 jaar.

De automatische corrector (zie verderop) kan deze ontleding en de bijbehorende foutcode vervolgens gebruiken bij het voorstellen van verbeteringen. In deze zin is zowel een verandering van *hij* in *ik* als een verandering van *word* in *wordt* mogelijk, maar gezien de aard van d/t-fouten gaat het systeem ervan uit dat waarschijnlijk de persoonsvorm fout is en niet het onderwerp.

3.3 Trifoonanalyse

Daelemans, Bakker en Schotel (1984) maken een onderscheid tussen twee strategieën voor correctie van tik- en spelfouten: een statistische en een linguïstische. Veel tekstverwerkers maken

gebruik van een statistische strategie. Dergelijke strategieën maken gebruik van het feit dat de ingetypte vorm weinig zal afwijken van de bedoelde. Ze corrigeren dan ook slechts typografische fouten. De spelling van het ingetypte woord wordt vergeleken met alle woorden in de woordenlijst en voor elk paar wordt een gelijkenismaat berekend. Vaak is deze strategie uitgebreid met statistische kennis over tikfouten, zoals het feit dat typografische fouten zelden aan het begin van een woord voorkomen. Het woord met de hoogste gelijkenismaat wordt vervolgens voorgesteld als correctie voor het ingetypte woord. Meestal geeft het systeem meerdere kandidaten waaruit de gebruiker dan zelf de bedoelde vorm kan kiezen.

Een linguïstische strategie is eerder gericht op de correctie van orthografische fouten. Deze strategie maakt gebruik van het feit dat de ingetypte woordvorm dezelfde uitspraak heeft als de bedoelde. Kennis over de relatie tussen uitspraak en spelling in een taal is hiervoor onontbeerlijk. Met behulp van deze kennis worden één of meerdere fonetische transcipties van het onbekende woord gemaakt. Die worden vervolgens opgezocht in een lijst met alle vooraf berekende en opgeslagen uitspraken van woorden in het lexicon. Als een uitspraak in de lijst overeenstemt met die van het ingevoerde woord, is de spelling van het woord in de lijst een mogelijke correctie. Met behulp van deze strategie kunnen ook enkele typografische fouten, bijvoorbeeld sommige letterverdubbelingen, gecorrigeerd worden, maar in de regel is deze methode niet geschikt voor correctie van typografische fouten.

Van Berkel en De Smedt (1988) stellen een combinatie voor van een statistische en een linguïstische strategie. Deze gecombineerde methode, die zij *trifoonanalyse* noemen, is gericht op de correctie van zowel typografische als orthografische fouten, zelfs wanneer beide typen in één woord voorkomen. Aangezien orthografische fouten moeilijker en persistenter zijn dan typografische, is als uitgangspunt een linguïstische strategie gekozen. Deze maakt een fonologische transcriptie van een foute woordvorm. De fonologische code wordt echter niet als zodanig opgezocht in een uitsprakenlijst, want dan zouden typografische fouten niet gecorrigeerd worden. Om de uitspraken op te zoeken wordt uitgegaan van *trigramanalyse*, een statistische strategie waarbij een woord opgedeeld wordt in overlappende segmenten van drie letters. Het woord *trigram* bijvoorbeeld wordt opgedeeld in *#tr, tri, rig, igr, gra, ram* en *am#* (waarbij # een spatie aangeeft). Alle trigrammen van een verkeerd gespeld woord worden opgezocht in een lijst van trigrammen met bijbehorende woorden waarin ze voorkomen. De woorden die op deze wijze het vaakst gevonden worden, zijn mogelijke correcties.

Wanneer trigramanalyse wordt toegepast op een fonologische transcriptie, spreken we van *trifoonanalyse.* De fonologische code wordt opgedeeld in trifonen, bestaande uit overlappende segmenten van drie fonemen. Deze trifonen worden vervolgens opgezocht in een lijst van trifonen met bijbehorende woorden waarin ze voorkomen. De woorden die zo het meest frequent gevonden worden, zijn mogelijke correcties. Deze methode werkt snel en goed, vooral voor spellingscorrectie van eigennamen en plaatsnamen. Deze leiden immers vaak tot zeer afwijkende spellingen die onvoldoende overeenkomen met de bedoelde om alleen op spelling gevonden te worden. Vergelijk de juiste spelling van *Stuttgart* en de uitspraak die ongeveer overeenkomt met *Sjtoedkard*. Naarmate het woord korter is, werkt trifoonanalyse, net als andere correctiemethoden, minder goed. Het woord is dan als geheel minder herkenbaar. Figuur 2 geeft een schematisch overzicht van het correctieproces voor de foutieve spelling *stylits* (een combinatie van een orthografische en een typografische fout) i.p.v. *stilist.*

1. fonologische transcriptie: *stylits → #stilIts#*
2. bepaling van trifonen: *#st, sti, til, ilI, lIt, Its, ts#*
3. opzoeken van trifonen in lijst, vinden van bijbehorende spellingen:

#st → staan, stem, stilist, ...
sti → elastiek, stiel, stilist, ...
til → stilist, tactiel, Tiel, ...
...
ts# → klots, muts, poets, ...

4. bepalen van de vaakst gevonden woorden: *stilist*

Figuur 2: Trifoonanalyse voor de spelling stylits.

Omdat deze correctiemethode, net als elke andere correctiemethode, vaak meerdere kandidaten voor correctie oplevert, kan hij verder worden verfijnd. Een in ons systeem toegepaste verfijning is om de gevonden woorden te vergelijken met het ingetikte woord op een aantal andere criteria, bijvoorbeeld woordlengte, en aldus een rangorde te bepalen. Een andere reeds genoemde verfijning is het filteren van de kandidaten door middel van informatie over de syntactische of semantische context. Het totale aantal voorstellen voor verbetering vermindert daardoor vaak drastisch.

3.4 Grammaticaspreadsheet en correctie van incongruentie

De functies van het grammaticaspreadsheet en de correctie van incongruentie worden beide uitgevoerd door een component die *propagatie* verricht. Propagatie is het doorgeven van syntactische kenmerken als *getal*, *genus* en *persoon* naar de woorden waarmee ze moeten overeenkomen. In Tabel 2 zijn de verschillende vormen van congruentie weergegeven die thans in de auteursomgeving zijn gedefinieerd.

Tabel 2: Enkele congruentierelaties.

A. Syntactische congruentie

1. Onderwerp met persoonsvorm *(persoon en getal)*
   Voorbeelden:
   Hij heeft een computer → Zij hebben een computer *(getal)*
   Hij heeft een computer → Ik heb een computer *(persoon)*
2. hoofd van NC met lidwoorden, voornaamwoorden, adjectieven en deelwoorden als voorbepaling *(genus, definietheid, getal)*
   Voorbeelden:
   De kleine computer → Het kleine computertje *(genus)*
   Het kleine computertje → Een klein computertje *(definietheid)*
   Het kleine computertje → De kleine computertjes *(getal)*

B. Semantische congruentie

Coreferentiële zelfstandige naamwoorden of persoonlijke voornaamwoorden *(geslacht en getal)*

Voorbeelden:

De jongen heeft zijn computer besteld → Het meisje heeft haar computer besteld *(geslacht, niet genus!)*

De jongen heeft zijn computer besteld → De jongens hebben hun computer besteld *(getal)*

Het doorgeven van kenmerken verloopt in één richting. Bij verandering van een woord via het grammaticaspreadsheet worden de kenmerken doorberekend vanuit het veranderde woord zelf. Bij correctie is niet zonder meer duidelijk welk woord bepalend is voor de overeenkomst. Daarom wordt de vuistregel gehanteerd, dat binnen de zin het onderwerp de bepalende constituent is en binnen de nominale constituent het zelfstandig naamwoord of persoonlijk voornaamwoord dat hoofd is van de constituent. Bij zin (12) wordt daarom correctie (12') verkozen boven (12").

(12)* Ik kan u melden dat de stukken gisteren verstuurd is.
(12') Ik kan u melden dat de stukken gisteren verstuurd zijn.
(12") Ik kan u melden dat het stuk gisteren verstuurd is.

Congruentie vindt echter niet alleen op syntactische gronden plaats, maar ook op semantische. Zo hebben *ik* en *mijn* in zin (13) dezelfde kenmerken, omdat ze coreferentieel zijn. Bij een verandering van een woord in een coreferentiële relatie, bijvoorbeeld de wijziging van (13) in (13'), moeten de coreferentiële woorden aangepast worden. Coreferentie is echter niet zonder meer voorspelbaar: zowel zinnen (13') als (13") zijn immers mogelijk. Daarom berust de beslissing over coreferentie in principe alleen bij de auteur.

(13) Ik heb mijn auto verkocht.
(13') Wij hebben onze auto verkocht.
(13") Wij hebben mijn auto verkocht.

Het aangeven van coreferentiële verbanden door de auteur is tijdrovend, maar het is slechts éénmaal nodig. Daarna kunnen zij opgeslagen worden en zij blijven dan bekend, niet alleen voor aanbrengen van wijzigingen via het spreadsheet, maar ook bijvoorbeeld voor vertaling. Congruentie tussen coreferentiële constituenten kan over de zinsgrens heen gaan. Het grammaticaspreadsheet kan er in dat geval op eenvoudige wijze zorg voor dragen dat hele teksten syntactisch worden aangepast waar nodig. Zo kan een hele tekst die in de *ik-vorm* staat naar de *wij-vorm* omgezet worden. Het grammaticaspreadsheet zorgt er dan voor dat persoonsvormen de nodige aanpassingen ondergaan, en ook dat bijv. *mij* in *ons* verandert en *mijn* in *ons* of *onze.*

3.5 Lettergreepscheiding

In de auteursomgeving kan lettergreepscheiding behulpzaam zijn, bijvoorbeeld bij het zo fraai mogelijk uitvullen van een bladzijde. Letterergreepscheiding is nagenoeg onmisbaar wanneer teksten in smalle kolommen worden gezet, bijv. in kranten. Het splitsen van een woord in lettergrepen brengt evenwel problemen met zich mee: hoe kan de perfecte splitsing bepaald worden? Problemen van semantische aard (zoals het onderscheid tussen *kwarts-lagen* en *kwart-slagen*) buiten beschouwing gelaten, is een goede methode mogelijk. Daelemans (1987) beschrijft een algoritme voor lettergreepscheiding op morfologische en fonologische basis. Deze methode bestaat uit twee stappen. Ten eerste worden de interne woordgrenzen bepaald. Deze zijn namelijk altijd mogelijke afbreekposities. Vervolgens worden binnen elk morfeem afzonderlijk de lettergrepen bepaald.

Het opdelen van het woord kan op vele manieren gebeuren. De methode van Daelemans tracht een woord te splitsen door een zo lang mogelijk woord van achteren van de samenstelling af te halen, en dit vervolgens te herhalen op het resterende gedeelte. Hierbij maakt de methode gebruik van een lijst van woordvormen met informatie over de interne woordgrenzen (bijvoorbeeld *in#entings#papieren*). Bij het splitsen van nieuwe samenstellingen, die niet in het lexicon voorkomen, worden de woordvormingsregels in acht genomen. Deze regels controleren de grammaticaliteit van de samenstelling. Zo kan bijvoorbeeld geen andere vorm van een werkwoord behalve de stam als niet-laatste deel van een samenstelling voorkomen (bijvoorbeeld *afspeelautomaat*). Tevens controleert de methode het gebruik van de bindmorfemen voor zover daar morfologische regels voor zijn: *stadswal* wordt bijvoorbeeld wel goedgekeurd, maar *stedenswal* niet, omdat het bindmorfeem *s* alleen na enkelvoudige zelfstandige naamwoorden voorkomt.

Het splitsen van een morfeem in lettergrepen geschiedt door het morfeem eerst in clusters van klinkers en medeklinkers te scheiden. Daarna worden de volgende regels toegepast op de medeklinkerclusters:

1 Als een cluster uit slechts één medeklinker bestaat, valt de lettergreepscheiding vóór deze cluster, bijvoorbeeld *he-ren.*
2 Als een cluster uit twee medeklinkers bestaat, valt de grens tussen deze medeklinkers (*wer-ken*); uitzonderingen hierop zijn de zogenaamde *cohesieve* clusters (zoals *ph* en *vr*): bij deze valt de grens vóór de cluster, dus *li-vrei.*
3 Als een cluster uit 3 of meer medeklinkers bestaat, valt de grens vóór het langst mogelijke cluster dat nog als begin van een Nederlandse lettergreep kan voorkomen (bijvoorbeeld *bar-sten* en *herf-stig*).

Binnen de klinkerclusters wordt op basis van spellingregels van voor naar achter gezocht naar het langst mogelijke segment; *aaien* wordt dus gesplitst in *aai-en* en niet in *aa-ien.* Ook binnen klinkerclusters zijn er uitzonderingen: zo blijft *ie* doorgaans in dezelfde lettergreep, behalve in *iee,* zoals in *financieel,* waar de grens tussen *i* en *e* ligt.

De hier beschreven methode werkt zeer goed. Zelfs zonder gebruik te maken van de kennis omtrent woordgrenzen, dus zonder lexicon, bepaalt het algoritme in meer dan 99% van alle woorden in lopende tekst de juiste scheiding. De fouten die nog gemaakt worden zijn te verhelpen door toevoeging van woordgrensinformatie. Slechts een uiterst klein deel van alle woorden wordt dan nog fout gesplitst; het betreft hier samenstellingen waarin de woordgrenzen niet eenduidig vastliggen, zoals in *kwartslagen* en *zeefriet.*

4 Toekomstige uitbreidingen

4.1 De Schooltekstverwerker

Gebruikers van een auteursomgeving zijn vooral geïnteresseerd in een zo snel en efficiënt mogelijke *correctie* van álle fouten. Het prototype van de auteursomgeving is dan ook aan deze wensen aangepast. De faciliteiten van de auteursomgeving zijn ook zeer nuttig voor een andere groep gebruikers: leerlingen die nog moeten leren schrijven. Voor deze doelgroep komen echter *detectie* en *verklaring* van de fouten op de eerste plaats.

Het schrijfonderwijs heeft tot doel leerlingen teksten (opstellen, brieven, essays, scripties) te leren schrijven, die qua inhoud, argumentstructuur en stijl zodanig opgebouwd zijn, dat zij voor een bepaalde doelgroep duidelijk en prettig leesbaar zijn. Belangrijk hierbij zijn correcte spelling en zinsbouw. Dit technisch correct schrijven is een vaardigheid die leerlingen mede via grammatica- en spellingonderwijs leren beheersen. In de praktijk blijkt deze vaardigheid echter nauwelijks: teksten van leerlingen bevatten veel technische schrijffouten. Dit heeft tot gevolg dat

1 de leraar zeer veel tijd kwijt is met het aanstrepen van grammatische fouten en spelfouten;
2 het oordeel van de leraar over de andere aspecten van de tekst negatief beïnvloed wordt (Van Oudenhoven, Withag en Siero, 1984);
3 de leraar nauwelijks toekomt aan de beoordeling van die andere aspecten;
4 de leerling zodanig gedemotiveerd raakt dat hij zijn fouten niet meer zorgvuldig bestudeert en verbetert;
5 de leerling weerzin ontwikkelt tegenover schrijven vanwege de technische problemen.

Het gebruik van een willekeurige tekstverwerker in het schrijfonderwijs heeft het grote voordeel dat de leerling—zonder een hele tekst te hoeven overschrijven—het commentaar dat de leraar *achteraf* geeft op een eenvoudige wijze in de tekst kan verwerken, o.i. de enige wijze om werkelijk nut te hebben van de aanwijzingen (zie bijvoorbeeld Looijmans en Schrauwen, 1986). Uitbreiding van de tekstverwerker met taalkundige faciliteiten zoals die boven beschreven zijn, geeft de leerling *vooraf* de mogelijkheid een aantal technische schrijffouten te corrigeren. In het kader van het SVO-project 5630 hebben wij een prototype ontwikkeld van een dergelijke tekstverwerker: de *Schooltekstverwerker* (Vosse, 1988).

De Schooltekstverwerker is globaal uit dezelfde modulen opgebouwd als de auteursomgeving. Hij kan dan ook dezelfde fouten detecteren als deze. Bij het ontwerp hebben wij echter een aantal specifieke beslissingen moeten nemen met het oog op toepassing in het onderwijs. Het betreft hier beslissingen op het gebied van bediening, lexicon, detectie en correctie, en feedback.

Dat een leerling die af en toe een tekst schrijft met behulp van een tekstverwerker heel andere eisen stelt aan de bediening en de mogelijkheden ervan dan een professionele typist, zal iedereen duidelijk zijn. De Schooltekstverwerker bevat dan ook maar een zeer beperkt aantal commando's, die afgeleid zijn van Nederlandstalige opdrachten als *knip, plak* en *controleer zin.* Verder wordt gebruik gemaakt van menuutjes, waarin de leerling door middel van de pijltjestoetsen zijn keuze kan aanwijzen. Ook worden speciale eisen gesteld aan de omvang van het *lexicon*. Een te groot woordenboek kan leiden tot het goedkeuren van vormen die de leerlingen—gezien hun woordenschat—zeker niet bedoeld kunnen hebben. Bij een te klein woordenboek zal het systeem veel goede vormen niet herkennen, waardoor de leerlingen te vaak ten onrechte een foutmelding krijgen. Wij hebben daarom in de Schooltekstverwerker een woordenboek opgenomen dat de tienduizend meest frequente woorden van het Nederlands bevat, met alle daarvan afgeleide woordvormen: de *Top Tienduizend.* Het programma kan op verzoek van de leerling ook tijdelijk andere woorden onthouden.

Op het gebied van de *detectie* van fouten moesten wij kiezen of het systeem als het ware over de schouder van de leerling zou meekijken (en dus meteen iedere fout zou signaleren), of dat het zou wachten op een expliciet verzoek van de leerling. Wij hebben voor de laatste optie gekozen, omdat een leerling niet voortdurend gestoord mag worden bij het conceptualiseren en formuleren, en omdat de leerling ook de kans moet hebben zelf zijn fouten te verbeteren zonder de hulp van het systeem. De leerling kan het systeem op ieder willekeurig moment controles laten uitvoeren. In zoverre wijkt de Schooltekstverwerker niet af van de auteursomgeving. Op het

gebied van *correctie* is er wel enig verschil. Gezien het didactische doel van de Schooltekstverwerker moet de leerling zelf zijn fouten corrigeren: hij moet immers leren deze in het vervolg te vermijden. Automatische correctie is dus niet mogelijk. Wel kan de leerling vragen om een menu met suggesties voor verbetering.

In het ideale geval zou de Schooltekstverwerker alleen die fouten moeten melden, die de leerlingen op basis van hun kennis van grammatica en spelling niet meer zouden mogen maken, en die zij dus ook zelf kunnen corrigeren. Hiertoe zou de Schooltekstverwerker gekoppeld moeten worden aan de onderwijsprogramma's voor grammatica en spelling die wij ontwikkelen in het kader van SVO-project 5620 (zie Pijls, Daelemans en Kempen, 1987). In deze programma's werken de leerlingen met boomstructuren, net als in de auteursomgeving. Deze bomen zijn echter volledig aangepast aan het kennisniveau van de leerlingen. Koppeling van beide programma's zou ook het gebruik van (aangepaste) bomen in de Schooltekstverwerker mogelijk maken. Bovendien zouden de leerlingen dan, bij het herhaaldelijk maken van dezelfde fout, verwezen kunnen worden naar de betreffende leerstof.

De Schooltekstverwerker is tot nu toe alleen geïmplementeerd als prototype op een computer die financieel ver buiten het bereik van scholen ligt. Desondanks zijn de voordelen ervan duidelijk te zien. Voordelen voor de leerlingen, doordat zij gemotiveerd worden hun kennis van grammatica en spelling te gebruiken, doordat hun cijfers niet negatief beïnvloed worden door technische schrijffouten, maar vooral doordat zij op het eigenlijke schrijfprodukt—op inhoud, argumentstructuur en stijl—feedback zullen krijgen die zij op eenvoudige wijze in hun teksten kunnen verwerken. En voordelen voor de leraar, die een deel van het werk uit handen genomen wordt en die dichter bij de verwezelijking van zijn lesdoelen kan komen.

4.2 Generatie van semi-standaardteksten

Een groot deel van zakelijke correspondentie bestaat uit standaardbrieven, die ongewijzigd naar meerdere ontvangers worden gestuurd, en semi-standaardbrieven, die enigszins worden aangepast aan de ontvanger ("open plaatsen correspondentie") of worden samengesteld door een selectie te maken uit standaardonderdelen ("bouwsteencorrespondentie"). Ook contracten, notariële acten e.d. zijn semi-standaarddocumenten.

Thans worden vele van deze teksten met behulp van een computer aangemaakt op een vrij omslachtige manier. Een zogenaamde *open-plaatsen*-tekst is opgeslagen als een raamwerk waarbinnen enkele variabelen zijn aangebracht. De auteur kiest een raamwerk en vult de variabelen één voor één in. Deze variabelen kunnen slechts volgens vastliggende patronen ingevuld worden. De consequenties van een bepaalde invulling voor de gehele tekst zijn niet taalkundig gedefinieerd en voor de gebruiker onbekend. Een brief die bijvoorbeeld gericht is aan één persoon, kan niet gemakkelijk veranderd worden in een brief aan meerdere personen. Hiervoor is een nieuw raamwerk noodzakelijk. Een ander nadeel van deze systemen is dat alle variabelen vooraf bedacht moeten zijn. Deze systemen zijn dus weinig flexibel.

Bij *bouwsteen*correspondentie doet zich een vergelijkbaar probleem voor. Alle bouwstenen worden moeten vooraf bedacht en opgeslagen zijn in het systeem. Een tekst wordt samengesteld door codes van aparte bouwstenen in te voeren. Doordat deze bouwstenen niet taalkundig gerelateerd zijn moet de keuze tussen bijvoorbeeld enkelvoud of meervoud moet bij iedere bouwsteen weer opnieuw gemaakt worden. Ook hier is het niet mogelijk om een tekst te veranderen nadat hij is samengesteld. Dit probleem is geworteld in het feit dat aan de

inhoudelijke keuze van een onderdeel impliciet ook een concrete realisatie, en dus een syntactische keuze, is verbonden.

Een auteursomgeving met een grammaticaspreadsheet maakt een veel flexibeler systeem mogelijk. Een dergelijk raamwerk voor semi-standaardbrieven gaat uit van prototypische bouwstenen waarvan de syntactische analyse en de coreferentiële verbanden door het systeem gekend zijn. Het aanpassen van een semi-standaarddocument kan dan gebeuren door het vervangen van elementen in de tekst zelf via het grammaticaspreadsheet (bijvoorbeeld *ik* vervangen door *wij*, *u* door *jij*). Deze veranderingen worden dan gepropageerd door het hele document naar constituenten die een relatie van coreferentie of congruentie hebben met de veranderde elementen. Deze manier van werken scheidt inhoudelijke keuzen van syntactische keuzen en laat ook niet voorziene varianten toe. Als bijkomend voordeel heeft de auteur steeds een concrete tekst als een prototype voor zich.

5 Slotopmerkingen

In het voorgaande hebben wij een ontwerp getoond van een auteursomgeving, waarin een auteur taalkundige ondersteuning krijgt bij het schrijven van een tekst. Dergelijke systemen kunnen worden ingezet in een kantooromgeving of in een onderwijssituatie. Zij kunnen de kwaliteit van de spelling en zinsbouw enigszins verhogen wanneer geen tijd of geld beschikbaar is voor professionele menselijke correctie. Het huidige prototype van de auteursomgeving is vooral gericht op spelling (ook binnen het zinsverband) en andere operaties op woorden zoals lettergreepscheiding en raadplegen van een lexicon. Naast taken op het woordniveau zijn er in de toekomst ook nog andere taken weggelegd voor auteursomgevingen. Met name op het gebied van stijl, woordkeuze, woordvolgorde, retorische structuur en begrijpelijkheid kan ons huidig prototype nog geen bijdrage leveren. Om correctie op deze gebieden behoorlijk uit te voeren is het meestal noodzakelijk dat de computer de teksten ook begrijpt, hetgeen bij de huidige stand van zaken nog niet mogelijk is.

Tekstverwerkers met taalkennis profiteren van vorderingen op gebied van o.a. de computertaalkunde, de computationele psycholinguïstiek, de computationele lexicografie, de informatica en de cognitieve ergonomie. Op hun beurt kunnen auteursomgevingen informatie leveren aan deelgebieden van de taal- en tekstwetenschap. Door bijvoorbeeld de in teksten gedetecteerde taalfouten automatisch op te slaan en deze vervolgens te ordenen volgens taalkundige principes, zou men een corpus van taalfouten kunnen aanleggen waar de taalpsychologie en het taalonderwijs uit kunnen putten.

---

[1] De term *grammatisch* duidt op ‘zaken die betrekking hebben op de grammatica’. De term *grammaticaal* wordt gehanteerd als zijnde ‘in overeenstemming met de regels van de taal’.

[2] Wij gebruiken de term *congruentie* hier in ruime zin, d.w.z. niet alleen voor de overeenkomst tussen onderwerp en persoonsvorm, maar ook bijvoorbeeld voor de overeenkomst tussen betrekkelijk voornaamwoord en antecedent, zelfstandig naamwoord en bijvoeglijk naamwoord als voorbepaling, etc.

Referenties

Berkel, B. van en K. De Smedt 1988, Triphone analysis: a combined method for the correction of orthographical and typographical errors. *Proceedings of the second conference on applied natural language processing, Austin, TX*. Association for Computational Linguistics, 77-83.

Daelemans, W. 1987, *Studies in Language Technology: an Object Oriented Computer Model of Morphophonological Aspects of Dutch.* Dissertatie Katholieke Universiteit Leuven.

Daelemans, W., D. Bakker en H. Schotel 1984, Automatische detectie en correctie van spelfouten. *Informatie*, 26, 949-1024.

Kempen, G., G. Anbeek, P. Desain, L. Konst en K. De Smedt 1987, Auteursomgevingen: tekstverwerkers van de vijfde generatie. *Informatie*, 29, 988-993.

Konst, L. 1986, *A syntactic parser based on filtering.* Intern rapport voor ESPRIT project OS-82, Psychologisch Laboratorium, Universiteit van Nijmegen.

Linden, E. van der en K. De Smedt 1987, Computerlexica voor een auteursysteem. *Toegepaste taalwetenschap in artikelen*, 27, 33-41.

Looijmans, P. en D. Schrauwen 1986, Een schrijfcursus via Alexis: teamwork van docent en computer (1). *Tijdschrift voor Taalbeheersing*, 1986, 1, 24-41.

Oudenhoven, J. van, J. Withag en F. Siero 1984, De invloed van spelling- en grammaticale fouten op de beoordeling van taalprestaties van leerlingen uit verschillende sociale milieus. *Nederlands Tijdschrift voor de Psychologie*, 39, 61-72.

Pijls, F., W. Daelemans en G. Kempen 1987, Artificial Intelligence tools for grammar and spelling instruction. *Instructional Science*, 16, 319-336.

Richardson, S. en L. Braden-Harder 1988, The experience of developing a large-scale natural language text processing system: CRITIQUE. *Proceedings of the second conference on applied natural language processing, Austin, TX*. Association for Computational Linguistics, 195-202.

Vosse, T. 1988, *Een slimme tekstverwerker voor grammaticale ondersteuning van het schrijfonderwijs*. Lezing ORD 1988, Leuven. Te verschijnen in F. Pijls en J. Sandberg (red.), nog zonder titel. Muiden: Coutinho.

Summary

The field of natural language processing has been dominated by machine translation and question answering systems. Other areas where the application of linguistic knowledge could be useful, such as word processing, have largely been ignored. In this article, we describe an editorial support environment *which offers linguistic support to authors while they are creating or editing documents. We discuss automatic correction of typing and spelling errors (including grammatical errors such as d/t confusion in Dutch verbal inflexions), reliable hyphenation, and consultation of an electronic dictionary. We also present some extensions: an educational word processor and a generator of semi-standard documents.*

Over de auteurs

Koenraad de Smedt is taalkundige. Als universitair docent aan de Universiteit van Nijmegen doceert hij in de studierichting Cognitiewetenschap. Hij verricht onderzoek op het gebied van de computationele psycholinguïstiek bij het Nijmeegs Instituut voor Cognitieonderzoek en Informatietechnologie.

Carla Huls is psychologe. Als junior wetenschappelijk onderzoeker aan het Nijmeegs Instituut voor Cognitieonderzoek en Informatietechnologie doet zij onderzoek o.a. op het gebied van auteursomgevingen in het kader van ESPRIT-project OS-82.

Fieny Pijls is taalkundige. Zij is projectleider van de SVO-projecten 5620 en 5630 voor het ontwikkelen van intelligente software ten behoeve van het taalonderwijs. Dit onderzoek wordt uitgevoerd aan het Nijmeegs Instituut voor Cognitieonderzoek en Informatietechnologie.