Het volgende woord

Rede, uitgesproken bij de openbare aanvaarding van het ambt van hoogleraar Geheugen, Taal en Betekenis aan de Universiteit van Tilburg op 10 oktober 2008 door dr. Antal van den Bosch.

Mijnheer de rector magnificus, geachte aanwezigen,

Ik prijs me gelukkig dat ik, in het bijzijn van de mensen die er voor mij toe doen, hier mijn ambt mag aanvaarden met het uitspreken van de hierna volgende reeks van 4.758 woorden.

1 Het probleem van het volgende woord

Woorden zijn net groepsdieren. In lange rijen zien we ze voorbij trekken op ons netvlies, en trillen hun klanken tegen onze trommelvliezen. Doorgaans komen ze netjes een voor een. De kudde die we langs zien trekken bevat niet allemaal unieke exemplaren. Sommige woorden, het zijn meestal de kleintjes, rennen onmiddellijk na hun passage achter ons terug om weer in de rij aan te sluiten. Als we wat langer waarnemen, blijken ook de andere, vaak langere woorden de neiging te hebben om terug in de rij aan te sluiten, maar ze doen er wat langer over. Meer in het algemeen kunnen we vaststellen dat vrijwel ieder woord vroeg of laat een plaatsje terugzoekt in de rij.

Een taalkundige zal je kunnen vertellen dat er een tweedeling te maken is in de zogenaamde functiewoorden, de kwikzilverachtige exemplaren die steeds maar weer terugrennen, en de inhoudswoorden, die vanwege het moeten meezeulen van een belangwekkende lading, die ook wel eens "betekenis" wordt genoemd, pas weer terugkeren als dat echt nodig is. Of, met een beetje meer fantasie zouden de functiewoorden gezien kunnen worden als ijverige lakeien, wiens functie het is om de entree van een of meer edele personages, de inhoudswoorden, in de hofzaal van het discours aan te kondigen.

De fantasie die ik hier beschrijf is een model van taal. De metafoor is wat barok. Toen de nieuwe wetenschap in de barok geboren werd was men dol op mechanische metaforen. Mechanieken zijn een goede opstap naar wiskundige en berekenbare modellen, bijvoorbeeld om er ook computermodellen van te maken. Ik ga dan ook de woordenrij mechanisch en iets abstracter voorstellen, en ik laat me assisteren door het beeld; beelden zeggen immers soms meer dan duizend woorden, en meer dan 4.758 moesten het er vandaag maar niet worden.

1.1 Een eenvoudig unigram-model

Een eindeloos lang stuk papier, het "medium", wordt gestaag afgerold. Het papier loopt door een mechaniek dat kan stempelen. Door het mechaniek heen lopen cirkelvormige banen. Iedere baan bevat een stempel met een woord. We beginnen met een enkel woord. Iedere keer als het woord het stempelmechanisme passeert, wordt het woord op het papier gedrukt.

Ieder woord heeft zijn eigen ring. Alle woorden bewegen met een constante snelheid. De kleine ringen draaien snelle rondjes met functiewoord-stempels, en op de langere ringen draaien de inhoudswoord-stempels geduldig hun rondjes. Zo kan het gebeuren dat er op een gegeven ogenblik een drietal woorden achter elkaar gestempeld worden.

Is dit een goed model van taal? Nee. De poging was aardig, maar dit eenvoudige mechanistische model (een zogenaamd unigrammodel) faalt in het genereren van echte taal. Als we het model vrij laten lopen, met een ring voor ieder woord, dan komen er reeksen uit als de volgende. Leestekens worden ook automatisch gegenereerd.

anders strijdmiddel te vinden Adams.

erbaasd voelde maart onze had het blijft 142 het in een willen De zeggen niet kwade kom elkaar popmuziek.

door nobelheid wisselkoersen.

betalen worden blinden Postma Tot komt proberen machtsovernames van 'personalia - sociaal deze het.

ja In behandeling ebolavirus stapte de de bij zijn schijnt verhullen is de in expres beeld nog Nederland ik.

Het is eenvoudig om te zien wat er mis is. In het echt lopen woorden niet willekeurig achter elkaar. Woorden horen bij elkaar omdat ze samen, in groepjes, iets te vertellen hebben. Soms komen ze in hun eentje, geheel onverwacht en lang niet meer of nog nooit gezien, en in dat geval hebben ze altijd iets belangrijks over te brengen. Soms komt een hele groep langs zonder veel te zeggen te hebben, zoals de vier slungels van "ik heb zoiets van". Ze komen in groepjes van twee, zoals "zeg eens", of "te voet", of in groepjes van drie, zoals "weet je nog" of "ik houd van".

De meeste woorden zijn lid van meerdere groepjes tegelijk. "Zeg" uit "zeg eens" zit ook in "kom nou zeg", en "ik zeg maar zo". Eigenlijk hoef ik dit niet uit te leggen; we kennen dit fenomeen allemaal wel, en belangrijker nog, we kennen de groepjes allemaal wel. Toch staan deze groepjes voor een groot deel niet in het woordenboek, want dat draait om woorden.

1.2 Een geheugen-gebaseerd woordvoorspellingsmodel

Het model met de ringen en stempels gaat tekort schieten als we ook willen dat woorden elkaar opzoeken, want er zijn teveel afhankelijkheden tussen teveel woorden. Als we bijvoorbeeld de ringen met elkaar zouden gaan verbinden, zodat het ene woord het andere woord zou meeslepen, dan zou het systeem vanwege de vele verbindingen direct vastlopen. Een ander, wat losser model is nodig. Hiervoor schakel ik over naar de praktijk van het onderzoek dat we hier in Tilburg doen, en wel naar een van onze eenvoudigste modellen van taal.

Dit model bouwen we op de computer, met behulp van een programma dat in grote hoeveelheden teksten zoekt naar groepen woorden die met elkaar in de rij willen staan. Dit computerprogramma wordt op pad gestuurd, de teksten in, met de opdracht op zoek te gaan naar rijtjes woorden die de volgende eigenschap hebben. Als je alle woorden in zo'n rijtje ziet behalve het laatste woord, dan kun je met aan zekerheid grenzende waarschijnlijkheid zeggen wat het laatste woord is. De rijtjes die het programma vindt, bestaan dus altijd uit twee woorden of meer; de meeste rijtjes zijn rond de drie of vier woorden lang. Om te beginnen laat ik een paar rijtje van twee woorden zien die zijn gevonden. Ik laat eerst het eerste woord zien. De vraag is steeds: wat is het volgende woord?

dezer — dagen
terugwerkende – kracht
aaneenschakeling — van
mijns — inziens

Dan een paar voorbeelden van rijtjes van drie of meer woorden.

maken deel — uit
graden onder — nul
heeft weten — te
tot een goed — einde
roet in het — eten
zicht terug te — trekken
leven in te — blazen
vers in het — geheugen

moet er niet aan — denken

Als de computer eenmaal dit soort rijtjes heeft gevonden, dan kan hij dus ook helpen bij het schrijven van teksten. Zodra we een volkomen voorspelbaar rijtje aan het intypen zijn kan de computer ons automatisch aanvullen, zodat we dat niet meer hoeven te typen. Dit is een voorbeeld van een aardig nut. Wat ook misschien aardig is, is dat we met deze verzameling rijtjes er een nieuw soort woordenboek voor het Nederlands bij hebben, alleen mag het natuurlijk geen woordenboek heten. Hoe dan wel? Daar kom ik later op terug.

Goed; ik was bezig met het bedenken van een model van taal dat het beter deed dan het mechanistische ringenmodel. Stel dat we de computer de opdracht geven om niet alleen naar rijtjes woorden te zoeken die voorspellend zijn voor één volgend woord, maar hem ook laten zoeken naar rijtjes waar twee of meer woorden op kunnen volgen. De hoeveelheid rijtjes die aan deze definitie voldoet is nog veel groter dan de verzameling rijtjes die we al hadden verzameld, maar deze opdracht is geen probleem voor een computer. We gaan nu dus ook opslaan in ons computermodel dat "is veel te" gevolgd kan worden door, onder andere, "gevaarlijk", "veel", "weinig","lastig", "duur", "mager", en "druk". Of dat "achterstallig" gevolgd kan worden door "onderhoud" of "loon". Achter me ziet u nog een paar van deze rijtjes. Ik stel in de kantlijn vast dat we ook deze rijtjes met meer dan één mogelijke volgende woorden grotendeels ook wel kennen.

Met de combinatie van beide soorten rijtjes in het geheugen kunnen we de computer op ieder gewenst moment in een tekst vragen wat het volgende woord zou moeten of kunnen zijn; de computer hoeft alleen maar de laatste paar woorden op te zoeken in zijn geheugen om te zien of ze bekend zijn als een rijtje dat typisch gevolgd wordt door een of meer woorden. Als het er één is, dan is dat de voorspelling; als het er meer zijn, dan kan de computer bijvoorbeeld het woord voorspellen dat het vaakst als laatste woord eindigde.

De computer heeft, met andere woorden, verwachtingen over hoe een Nederlandse zin op ieder willekeurig punt verder gaat. Het model, dat de vorm heeft van enkele honderdduizenden tot enkele miljoenen rijtjes, is daarmee een echt taalmodel geworden. Het is alleen niet een taalmodel zoals je dat uit het taalkundeboek kent. *It's linguistics, Jim, but not as we know it*.

1.3 De wet van Heaps

Hoe goed kan dit model eigenlijk voorspellen? Dat is op verschillende manieren te meten. We zijn begonnen dit in kaart te brengen door de percentages te meten van correct voorspelde woorden in teksten die de computer nog nooit had gezien, door de teksten woord voor woord te doorlopen, en steeds de beste gok van de computer te vergelijken met het echte volgende woord. We vonden tot nu toe allerlei antwoorden, zoals 6% correct voorspelde woorden, 15%, tot wel 50%. Maar bovenal hebben we een relatief antwoord gevonden: als de computer alsmaar meer tekst krijgt om rijtjes in te zoeken, dan vindt hij alsmaar meer rijtjes, en worden de voorspellingen alsmaar beter. We hebben dat relatieve antwoord ook bij benadering kunnen kwantificeren: iedere keer dat de hoeveelheid tekst met een vaste factor wordt verveelvoudigd, bijvoorbeeld vertienvoudigd, gaat het percentage correct voorspelde woorden met een redelijk constante hoeveelheid omhoog. Zo hebben we in een experiment met Engelse data met elke vertienvoudiging van het basis-tekstmateriaal een stijging van om en nabij de 8% gemeten. Het eindpunt van deze grafiek is het punt waarop we een grotere computer moeten zoeken.

Dit klinkt misschien net zo fabelachtig als de onwaarschijnlijke statistieken van Battus in zijn Opperlandse Taal- en Letterkunde. Het roept op z'n minst om een verklaring. Naarmate een grotere hoeveelheid teksten wordt genomen, komen steeds meer voorbeelden terug van woorden die eerder al eens voorkwamen—uiteraard vooral van de kleine groep functiewoorden, maar uiteindelijk komen alle woorden een keer terug. Tegelijkertijd verschijnen ook steeds meer nieuwe woorden. De information retrieval-onderzoeker H.S. Heaps beschreef in de jaren zeventig van de vorige eeuw hoe, als je steeds nieuwe teksten erbij neemt, je altijd weer nieuwe woorden blijft tegenkomen, maar deze groei neemt over de tijd heen wel af. Naarmate de computer meer

tekst ziet duurt het steeds langer voordat er zich weer een nieuw woord aandient. De groei van het aantal nieuwe woorden neemt dus af, maar is toch nog wel zo sterk dat bij iedere vertienvoudiging van de hoeveelheid tekst, het aantal nieuwe woorden met steeds grotere sprongen toeneemt. Heaps stelde de volgende vergelijking voor om deze groei te voorspellen. Twee variabelen zijn taalspecifiek. Voor het Engels geldt dat β meestal 0.5 is, en K rond de 50; n is het aantal woorden dat je tot nu toe gezien hebt, en V_R (de V van vocabulair) is het aantal unieke woorden dat je in die woordenmassa tot nu toe hebt gezien. En het klopt; in deze grafiek is te zien hoe de voorspelling van Heaps nogal goed overeenkomt met echte tellingen van unieke woorden in een verzameling van, in dit geval, Engelse journalistieke tekst, gemeten tot aan 50 miljoen woorden tekst. In 50 miljoen woorden Engelse tekst komen dus typisch ruim 500 duizend unieke woorden voor. Als de y-as ook logaritmisch zou zijn, zou de grafiek met een rechte lijn omhoog gaan.

Het dubbele effect van het alsmaar vaker zien van bekende woorden en het zien van steeds meer nieuwe woorden, kan er alleen maar toe leiden dat het herkennen van voorspelbare patronen alsmaar beter zal gaan, en daarmee ook het voorspellen van het volgende woord.

Ik meldde terloops dat het al eens gelukt is om boven de 50% correct voorspelde woorden uit te stijgen. De computer was voor die gelegenheid losgelaten op 60 miljoen woorden Engelse tekst; krantentekst om precies te zijn, uit de archieven van het persagentschap Reuters. Globaal gesproken worden de meest voorkomende woorden, de functiewoorden, het best voorspeld. Het passeren van de 50% roept meteen ook de vraag op wat er dan eigenlijk nog fout gaat. Wel, het zou niet verrassend moeten zijn dat het model vrijwel altijd fout zit met z'n voorspelling bij het begin van een tekst, en vaak ook bij het begin van nieuwe alinea's, zinnen of bijzinnen. Helderziend is het systeem nu ook weer niet.

Als het binnen in zinnen fout gaat, dan gaat het vaak mooi fout. In een recent experiment met het Nederlands wordt het woord "tweede" steevast incorrect voorspeld als "eerste". In plaats van "Brussel" en "Berlijn" voorspelt ditzelfde model een paar keer "Amsterdam". Het zijn maar anecdotische aanwijzingen van een algemeen patroon waarin het model op een volstrekt impliciete manier niet alleen rijtjes herkent en op een grammaticaal kloppende manier kan afmaken, maar ook vaak in ongeveer de juiste betekeniswolk aan het prikken is wanneer het een beste gok aan het doen is.

1.4 Het constructicon

Tijd voor een tussenbalans. Ik heb beschreven hoe uit grote hoeveelheden teksten rijtjes van woorden ontdekt kunnen worden, waarbij de eerste woorden het laatste woord voorspellen, of tenminste leiden tot een afgebakende groep mogelijke woorden. Al deze rijtjes bij elkaar vormen een soort woordenboek van het Nederlands, hoewel woordenboek dus niet de juiste term is. Om het toch een naam te geven, leen ik van Ad Backus en zijn collega's de term **constructicon**; lexicon van constructies. Dat er veel van zijn, is duidelijk; veel woorden komen in meerdere constructies voor. Eerder zei ik al dat we hier bij benadering spreken we hier over enkele honderdduizenden tot enkele miljoenen constructies; en daarbij vind je er meer naarmate je verder zoekt. Uitgeverijen zullen om die reden niet staan te springen om het Nederlandse Constructicon als boek op de markt te brengen. Toch hebben we het hier over belangrijke algemene taalkennis; pas als je weet hoe woorden met elkaar in de rij willen en kunnen staan, weet je "hoe je dingen zegt".

2 Toepassingen

Als niemand het Nederlandse Constructicon zal uitbrengen in boekvorm, wat hebben we er dan aan om het te bedenken? Nou, in zijn computervorm is er van alles mee te doen. Nuttige zaken; handige toepassingen. Ik laat twee voorbeelden zien van toepassingen die bepaalde aspecten van talige digitale communicatie, zoals tekstverwerking en internationale informatieuitwisseling wat gemakkelijker zouden kunnen maken.

Het eerste voorbeeld is het ontdekken van fouten in teksten. Iedereen die teksten schrijft, maakt fouten. Het overkomt ons allemaal. We maken die fouten liever niet, want hoewel ze ons

niet opvielen toen we ze maakten, merken anderen ze juist wel op. Marc van Oostendorp richtte een maand geleden in zijn oratie in Leiden zijn pijlen op het duivelse verschil tussen *d* en *t* in het Nederlands, maar verwees daarbij zijn toehoorders door naar mij als ze er ook een werkend computerprogramma bij wilde hebben. Daarom concentreer ik me hier op de de *dt*-fout.

2.1 De *dt*-fout

Ik hoef vermoedelijk niemand uit te leggen wat de bedoeling is van de *-t* uitgang van werkwoorden in het Nederlands. De regel is simpel, maar vergt een oplettendheid die zelfs door de meest geoefende schrijvers niet altijd is vol te houden. Die oplettendheid betreft vooral werkwoorden waarvan de stam op *d* eindigt. Het fundamenteel oneerlijke is dat het voor de uitspraak niet uitmaakt of zo'n werkwoord nu op *d* of op *dt* eindigt, maar wel voor de commissie die jouw sollicitatiebrief leest. Een "ik wordt" met *-dt* is zo geschreven. Volgens Google kun je "ik wordt" met *-dt* zo'n 315.000 keer vinden op internet. "Ik word" met een *d* komt met ruim anderhalf miljoen voorkomens tenminste nog vijf keer zo vaak voor; er is dus nog hoop. Die hoop kan bovendien aangewakkerd worden als we ons woordvoorspellende systeem voor dit probleem in stelling brengen.

De woordvoorspeller behoeft een paar kleine aanpassingen om er een *dt*-corrector van te maken. Allereerst stel ik de taak wat nauwer af: in plaats van het voorspellen van alle mogelijke woorden is hier de taak om te besluiten of het volgende woord eindigt op *-d* of *-dt*. Dit model hoeft ook minder vaak in actie te komen: we zoomen in op die woorden in een tekst waarvan we weten dat ze op *-d* of op *-dt* kunnen eindigen – de vraag is welke van de twee uitgangen op deze plek de juiste is. We hoeven hiervoor niet te weten wat werkwoorden zijn, of wat de regels zeggen over de *-t*-uitgang; we hoeven alleen maar paren van woorden te zoeken die hetzelfde zijn, behalve dat het ene woord eindigt op een *-d* en het andere op *-dt*. De laatste aanpassing is om de woorden die volgen op het *d-dt*-woord ook mee te nemen in de beslissing. Tot nu toe ging ik ervan uit dat het volgende woord nog niet bekend is, maar in een tekst die er al ligt en waar je de fouten uit wilt halen, zijn alle woorden bekend. Het probleem is nu juist: welk woord is fout?

Vervolgens zoeken we in een grote hoeveelheid tekst simpelweg alle voorkomens van *-d-* en *-dt*-woorden; dat worden de voorbeelden waarop we een zelfde soort model bouwen als eerder. We slaan rijtjes woorden op, nu links en rechts van het middelste *d-dt*-woord, waaruit je met grote waarschijnlijkheid kunt afleiden dat het woord in het midden een *-t*-uitgang krijgt, of niet. Het aantal rijtjes dat met deze procedure gevonden wordt passeert de anderhalf miljoen, en het eindresultaat is er ook naar: als we testen op nieuwe *d-dt*-woorden in teksten die het model nog niet eerder heeft gezien, dan maakt het model in 99,2% van de gevallen een correcte beslissing. Dat doet het op basis van typische *dt*-constructies als "Hij * dat de" en "Dat * straks" en met typische *d*-constructies als "ik * het", "* je", maar ook "het * van", want woorden als "geld" zijn in hun *d*-vorm niet alleen werkwoorden, maar ook zelfstandige naamwoorden. Dat weet ons model niet, maar dat geeft ook niet.

Met zo'n accuratesse, 99,2%, kan het model vervolgens losgelaten worden op iedere tekst waarin *d-dt*-woorden staat; ieder *d-dt*-woord wordt als mogelijke fout onderzocht. Wanneer het model iets anders voorspelt dan dat er feitelijk in de tekst staat, dan zou de fout wel eens aan de tekst kunnen liggen, en hebben we dus feitelijk een *dt* fout opgespoord. In gezamenlijk onderzoek heeft Herman Stehouwer het web afgespeurd naar *dt*-fouten, en heeft een bloemlezing gemaakt van enkele honderden gevallen. Het zal niemand verbazen dat het vrij eenvoudig is om op het web *dt*-fouten te vinden; ik gaf eerder al de cijfers voor "ik wordt" met *dt*. We vonden er bijvoorbeeld veel in teksten op forumpagina's, waarbij de zinnen doorgaans niet alleen maar lijden aan een *dt*-fout; meestal wordt de fout omringd door allerlei andere problemen. Ons model was in staat om 79% van die gevallen aan te duiden als fouten. Een redelijke score, zeker gezien de soms problematische gevallen als de onderste zin, waar de verkeerde vorm van het woord "zei" niet bepaalt helpt. Aangespoord door een redacteur van een bundel waarin dit resultaat zal worden gepubliceerd, hebben we ook een vergelijking gemaakt met de grammaticale correctiemodule die ingebouwd zit in Microsoft Office, versie 2003; deze bekende tekstverwerker merkte slechts 17% van de *dt*-fouten aan als fout.

Het *dt*-probleem is maar een fragment van het totale probleem van tekstcorrectie, en het zijn maar twee cijfers die ik hier vergelijk, maar met dit probleem zijn we al wel beland in het werk- en privé-domein van mensen zoals u en ik die regelmatig tekstverwerkers gebruiken. Het *dt*-probleem staat model voor heel veel meer gevallen van grammaticale en morfologische verwarbaarheid die op dezelfde manier aangepakt kunnen worden: *hen–hun*, *dan–als*, verwarringen op basis van klankgelijkenis, enzovoort. Het ligt in de mogelijkheden om tekstverwerkers van al deze "impliciete intelligentie" te voorzien; onze modellen kunnen er zo ingebouwd worden.

2.2 Automatisch vertalen

Een andere veelgehoorde wens waar onze woordvoorspeller een rol kan spelen, is de mogelijkheid om teksten te vertalen. De wereld wordt een stuk ruimer als je kunt lezen wat mensen in andere talen zeggen. Er is eigenlijk maar een eenvoudige aanpassing nodig om onze woordvoorspeller om te vormen in een automatische vertaler. Stel, we laten hem leren hoe je Nederlands naar Engels vertaalt. De taak is dan niet meer om een Nederlandse constructie af te maken met een volgend Nederlands woord, maar om die constructie te koppelen aan een Engels woord, of een Engelse constructie. Het model moet daarbij weten hoe je woorden en constructies in het Nederlands naar het Engels vertaalt, maar minstens zo belangrijk is dat het model weet hoe je dingen zegt in het Engels.

Er liggen nu een paar puzzelstukjes op mijn spreekwoordelijke onderzoekstafel die in elkaar lijken te passen. Bij vertalen hebben we het niet over hoe je dingen zegt in één taal, maar de simultane kennis over hoe je dingen zegt in twee talen. Die kennis kun je impliciet afleiden als je een heleboel vertalingen hebt. Bij het vertalen van "ik houd van sterke koffie", is het nodig om te weten dat "ik houd van" een typisch Nederlandse constructie is, en "sterke koffie" een andere; en dat "I love" en "strong coffee" typische Engelse constructies zijn, en, en dit is essentieel, dat "ik houd van" vertaalt naar "I love", en "sterke koffie" naar "strong coffee". Het probleem is dus niet om het woord "sterke" te vertalen, wat in principe vertaald zou kunnen worden als "strong" en "powerful". We hoeven ons dit niet af te vragen, want we zijn hier helemaal niet bezig met "sterke" te vertalen, maar met "sterke koffie", en dat heeft maar één vertaling: "strong coffee".

Hoe kan de computer overigens zeker weten dat "strong coffee" niet een vertaling is van "ik houd van"? Zekerheid is er nooit, maar als je een grote verzameling van vertalingen hebt, dan kun je door te tellen vaststellen dat de twee constructies "ik houd van" en "I love" heel vaak samen voorkomen in paren van vertaalde zinnen; in ieder geval vaker dan "ik houd van" en "strong coffee".

In de aanloop naar het bouwen van zo'n model, in gezamenlijk werk met Peter Berck en Sander Canisius, hebben we eigenlijk een nog simpeler model gebouwd, dat zich beperkt tot het omzetten van rijtjes van drie Nederlandse woorden naar rijtjes van drie Engelse woorden, waarbij de middelste woorden van beide rijtjes elkaars meest waarschijnlijke vertaling zijn. Vervolgens worden de Engelse rijtjes van drie op de goede volgorde gezet, door gebruik te maken van hun onderlinge overlap. Klinkt ingewikkeld misschien, maar dat is het niet. Het werkt zo.

We zitten in de constructiefase van het model, waarin we voorbeelden krijgen van vertaalde zinnen. We hebben voor één zo'n zin vastgesteld welke woorden elkaars vertaling zijn, waarschijnlijk (dat hebben we op dezelfde manier vastgesteld als ik eerder schetste met "sterke koffie"). We beginnen met de Nederlandse kant, en splitsen de zin op in losse rijtjes van drie woorden, waarbij ieder woord het midden is van één rijtje. Vervolgens koppelen we ieder Nederlands rijtje van drie aan een Engels rijtje, waarbij dus de middelste woorden elkaars vertaling zijn. Stel nu dat we deze zin willen vertalen, dan gaat dat perfect, want we hebben alle benodigde informatie in ons geheugen zitten. Het enige dat nog ontbreekt is de juiste woordvolgorde in de Engelse zin. Dat lossen we op door uit te gaan van de overlap tussen de rijtjes van drie, want die bevatten impliciet de informatie die je nodig hebt om de Engelse volgorde terug te krijgen. In het ideale geval, zoals bij deze zin die we als voorbeeld al kennen, kunnen we perfect de juiste Engelse zin herfabriceren.

Wat veel belangrijker is dan het terug kunnen genereren van vertalingen die je kent, is het vertalen van nieuwe zinnen. Het goede nieuws is dat de methode blijft werken, al gaat het al

snel niet meer zo perfect als in het voorbeeld. Het blijft grotendeels werken omdat voorspellingen van rijtjes van drie gedeeltelijk fout mogen zijn, zolang er maar een beetje overlap is tussen rijtjes (één woord is al genoeg). Met deze eenvoudige methode doen we het nog niet eens zo slecht. We doen het niet zo goed als Google Translate, de online vertaalmachine van Google die in grote lijnen dezelfde aanpak als de onze volgt, maar wel met constructies werkt, en over veel meer voorbeelden en veel meer grotere computers kan beschikken. De scores, gemeten met de zogenaamde BLEU-evaluatiemetriek, liggen tussen 0 en 100. Teruggerekend naar een cijfer tussen de 0 en de 10 halen we tenminste op wetsteksten een zes. Het klinkt naar zesjescultuur, maar ik durf toch te zeggen dat we hiermee best tevreden zijn, zeker gezien de radicale simpelheid van ons model.

Opvallend is dat we op een paar typen teksten, met name transcripten van sessies van het Europees Parlement, en Europese wetsteksten, met ons elementaire model al een bekend traditioneel systeem, Systran (ook wel bekend als Babelfish) voorbij zijn gestreefd. Dat komt vooral omdat wij ons model kunnen trainen op het typerende taalgebruik binnen zo'n domein, terwijl het 40-jaar oude Systran gemaakt is om zo algemeen mogelijk toepasbaar te zijn (wat onze systemen dus niet zijn), maar niet zomaar kan bijleren op basis van voorbeelden.

Ik laat een paar fouten van ons vertaalmodel zien. Hier volgen twee vertalingen van zinnen uit transcripten van het Europees Parlement. De voorbeelden tonen ook een wezenlijke moeilijkheid aan van het evalueren van een vertaling; net zo min als ons model helderziend is, kan het niet weten welke beslissing de vertaler heeft genomen uit de vele mogelijke vertalingen die een zin al snel kan hebben.

3 Implicaties

Ik ben taaltechnoloog, en als ik praat over mijn werk, zeker als dat met collega's is, dan lijkt dat al snel op een vergelijkende autotest in Top Gear. Hoewel ik altijd warm loop voor het optimaliseren van technische oplossingen, ligt mijn wezenlijke interesse ook op het vlak van de taalkunde. De indruk kan bestaan alsof ik de taalkunde in mijn werk zoveel mogelijk wegstop, en triomfantelijk roep: kijk eens wat ik kan zonder grammatica of betekenis! Maar ik doe het helemaal niet zonder grammatica of betekenis. Als onze modellen *dt*-fouten kunnen detecteren, dan bezitten ze grammaticale kennis, want het is een morfo-syntactisch probleem. Als mijn modellen kunnen vertalen, dan bezitten ze kennis van betekenis, want een succesvolle vertaling betekent ongeveer hetzelfde als het origineel. Uw vraag is: waar zit dan die kennis? Mijn antwoord is: die bevindt zich in het proces van analoog redeneren op basis van grote hoeveelheden voorbeelden.

Grammatica en betekenis als proces, het is even wennen misschien. In mijn ogen is het precies dat wat ons vakgebied bedoelt als het zich profileert met de naam "natural language processing", natuurlijke taal*verwerking*. Het gaat ons om het modelleren van de processen die plaatsvinden in taalgebruik; in het voeren van een dialoog, bij het vertalen, bij het omvormen van een tekst met fouten naar een correctere tekst, en ook in het beantwoorden van vragen, het samenvatten of parafraseren van tekst, of het uitspreken van tekst, om een aantal Tilburgse onderzoeksonderwerpen te noemen.

Een belangrijke eigenschap van mijn impliciete grammatica- en betekenismodellen is dat het geen black boxes, ondoorzichtige modellen zijn. Ze laten zich wel degelijk bekijken, doormidden snijden, en kwantitatief opmeten. Ze zijn alleen wel erg groot, met honderdduizenden tot miljoenen onderdelen, maar dat houdt vooral in dat er nog werk ligt in het vinden van goede analysemethoden om deze modellen beter te begrijpen. Er is in ieder geval weinig minimalistisch te ontdekken aan deze modellen; je zou ze eerder maximalistisch kunnen noemen. Maar wie zegt dat een model van taal minimalistisch moet zijn?

Met die retorische vraag resonerend in uw hoofden wil ik mijn bereidheid tonen om de discussie aan te gaan met iedereen die zich taalkundige noemt en voelt. Is het idee van grammatica en betekenis als proces, een werkbaar idee? Sluit het aan bij bestaande niet-computationele modellen van taal, en waar vloekt het mee? De tekenen zijn goed. In gesprekken met mijn Tilburgse collega's Ad Backus, Maria Mos, Seza Dogruoz en Anne Vermeer komen onze ideeën als

puzzelstukken bij elkaar. Walter Daelemans heeft in Antwerpen in de laatste vijftien jaar een brug kunnen slaan met de psycholinguistiek, bijvoorbeeld via het werk van Emmanuel Keuleers, waardoor onze modellen inmiddels ook gebruikt worden om menselijk taalgedrag te modelleren.

3.1 Een plan

Naast de buitenwaards gerichte blik is er natuurlijk de zorg voor het eigen bouwwerk. Daarin ben ik niet alleen; ik ben rijkelijk omringd door gelijkgestemden in de ILK onderzoeksgroep, en het Tilburg centre for Creative Computing. Vanuit deze fantastische inbedding wil ik in mijn ambt als hoogleraar toch zeker de volgende twee doelen voor ogen hebben en houden.

Ten eerste wil ik een nieuw taalkundig model presenteren: het impliciete taalkundige model, waar ik vandaag al over sprak. Hierbij sta ik op de schouders van generaties van wetenschappers die al sinds de jaren '50 bouwen aan analoog redenerende lerende systemen, in het bijzonder de ontwikkelaars van symbolische inductieve leertechnieken, bijvoorbeeld de instance-based learning algoritmes van David Aha en collega's. En om een andere inspiratiebron te noemen, ik gebruik ook de sterke schouders, net zoals Google dat doet, van ontwikkelaars van impliciete taalmodellen voor het terugvinden van documenten, ofwel information retrieval, een sterke discipline in Nederland, bovendien net nog verder versterkt met een nieuwe hoogleraar, Wessel Kraaij, in Nijmegen. Ik sta ook op de schouders van de ontwikkelaars van statistische taalmodellen, en statistische en exemplaar-gebaseerde modellen voor automatisch vertalen.

Deze ideeën bestaan dus allemaal al, maar onder vele verschillende namen. Tegelijkertijd is de equivalentie tussen al deze modellen eenvoudig te zien. Ze zijn allemaal gebaseerd op analoog redeneren tussen gememoriseerde en nieuwe voorbeelden. Het is niet mijn bedoeling om de mensen op wiens schouders ik sta uit te leggen dat ik een betere terminologie heb gevonden voor hun werk. Ik wil vooral naar buiten toe, naar de beschrijvende en toegepaste taalkunde, de psycholinguistiek, de sociolinguistiek, de diachrone taalkunde, en alle aanverwante vakgebieden binnen en buiten de ruime cirkel van taalstudies laten zien dat de computerlinguistiek, geboren uit het huwelijk van de taalkunde en de kunstmatige intelligentie, in een betrekkelijk zelfstandig proces van enkele decennia een breed inzetbaar model heeft geproduceerd waarin betekenis en syntax geen meta-talen zijn, maar processen.

Een voor mij heldere tweede ambitie is om dit alles te doen in een open forum, een marktplaats voor wetenschappelijk onderzoek, waarin technologie-ontwikkeling volgens de Open Source-filosofie hand in hand gaat met wetenschappelijke vraag-beantwoording. Ik kijk er ook naar uit om de samenwerking te verdiepen met de nieuwe groepen die we in Tilburg hebben verwelkomd binnen ons nieuwe Tilburg centre for Creative Computing, op het gebied van games en vision. Veel van wat ik hier heb gezegd over impliciete modellen heeft een evenknie in de wereld van gaming, robotica, en beeldverwerking; ook daar blijken maximalistische, analoog redenerende modellen goed toepasbaar.

Dankwoord

Aan het eind van mijn rede gekomen wil ik beginnen met mijn dank uit te spreken aan het college van bestuur van de Universiteit van Tilburg en het bestuur van de Faculteit der Geesteswetenschappen voor hun vertrouwen in mij. Een woord van dank wil ik ook richten aan NWO, de Nederlandse Organisatie voor Wetenschappelijk Onderzoek, voor het creëren van de voorwaarden om mijn ideeën op het gebied van onderzoek te realiseren.

Ik prijs mezelf gelukkig met de groep mensen die mij in dat onderzoek de weg hebben gewezen. De tijd die ik als student op deze universiteit doorbracht bracht me al vroeg op die weg. Ik herinner me de voelbare energie die, nu twintig jaar geleden, in de lucht hing rondom de start van een nieuwe doctoraalopleiding (Taal en Informatica) en een nieuw onderzoeksinstituut (het Instituut voor Taal- en Kennistechnologie), met vanuit ons studenten gezien als boegbeeld Harry Bunt. Hooggeleerde Bunt, beste Harry, je hebt structuren neergezet waarin wij ons hebben kunnen ontwikkelen. We werden meegenomen in het authentieke enthousiasme van mensen als

Hans Paijmans en Walter Daelemans, die niet alleen inhoud brachten, maar ons ook aanmoedigden om buiten de gebaande paden te denken. Buiten de deur leerde ik nog meer van Bea de Gelder, Jean Vroomen en René Collier. Ik ben mijn docenten van toen dankbaar voor het overbrengen van die energie en die zin in onderzoek doen.

Het was Walter Daelemans die met het onderwerp kwam waarover we een dik decennium jaar later samen een boek hebben geschreven. Het onderwerp, geheugengebaseerde taalverwerking, is een geweldige inspiratiebron gebleken waaruit we nog altijd putten. Hooggeleerde Daelemans, Walter, ik ben je dankbaar dat je me vanaf dat eerste moment deelgenoot hebt gemaakt van je ideeën. Je raadgevingen door de jaren heen hebben me op belangrijke momenten wezenlijk geholpen. Je hebt onze groep in Tilburg opgebouwd, parallel aan een zustergroep in Antwerpen. Ik hoop dat wat wij altijd gniffelend onze Groot-Brabantse gedachte noemen, nog lang moge bestaan. Jouw gastvrijheid en onze kameraadschap en vriendschap is me zeer dierbaar.

Het werk van Ton Weijters was een belangrijke inspiratiebron voor de eerste experimenten die Walter en ik uitvoerden. Later was hij ook een van mijn begeleiders van mijn promotiewerk. Ik dank Ton voor zijn vriendschap, toewijding, en wetenschappelijke eigenwijsheid, waar ik veel van heb geleerd.

Hooggeleerde Van den Herik, beste Jaap, hoe tomeloos je energie is ervaar ik inmiddels weer iedere dag. Met bewondering heb ik meegeholpen om een nieuwe droom te realiseren: een onderzoekscentrum voor Creative Computing in, hoe bestaat het, Tilburg. Maar toch klopt het helemaal: we zijn het altijd roerend eens gebleven over hoe goed het onderwerp "taal" past op jouw visie op het snijvlak van het menselijk blikveld en de computer. Ik bewonder je open blik over vakgebieden en specialismen heen. Ik kijk bijzonder uit naar de komende jaren van samenwerking binnen ons centrum.

Ik wil mijn dank ook uitspreken aan vier hooggeleerden die ik heb leren kennen toen ze nog niet die waardigheid bekleedden. Hooggeleerde Postma, Eric, je magische mix van rustgevend relativisme en waanzinnige humor hebben me in mijn promotietijd zeer geholpen. Met jou erbij komt altijd alles goed. Hooggeleerde Krahmer, beste Emiel, wat heb ik veel gehad aan je feedback in onze bijpraatsessies door de jaren heen. Onze gezamenlijke en parallele geschiedenis is nog niet afgelopen, en ik kijk uit naar het vervolg—en naar signaleringen van nieuwe muziektrends natuurlijk. Hooggeleerden Swerts en Maes, beste Marc en beste Fons, voor mij maken jullie deel uit van het Vlaamse engelenkoor wat over mij, kwart-Belg, altijd waakt. Ik prijs me gelukkig met jullie als collega's en dank jullie voor het luisterende oor en jullie support en aanstekelijke enthousiasme.

Ik werk dag in, dag uit samen met een groep mensen op wie ik enorm trots ben, en die ik diep dankbaar ben voor hun inzet en kameraadschap. Piroska, Martin, Toine, Ko, Sander, Sander, Steve, Marieke, Herman, Peter, Menno, Roser, Iris, Erwin, en alle anderen die ILK hebben helpen worden tot wat het nu is. Ik vind het fijn om met jullie te werken.

Mijn dank geldt ook, zonder namen te noemen, maar de goede verstaander weet genoeg, voor iedereen in Nederland en daarbuiten met wie ik heb mogen samenwerken, en nog steeds samenwerk, aan onderzoek en de organisatie van onderzoek.

Terug naar de basis. Ik haal veel kracht uit het gevoel van thuis zijn temidden van mijn gezin, waar ik ook ben. Dat gevoel, en die basis zelf, heb ik meegekregen van mijn lieve ouders. Lieve papa en mama, bedankt voor jullie nooit aflatende vertrouwen en steun; jullie kunnen ook op mij rekenen.

Lieve Liam en Merijn, wat is het toch mooi om jullie de wereld te zien betreden. Jullie originaliteit en fantasie verrast me iedere dag weer en maakt me gelukkig. En jullie worden zo groot. We gaan nog veel leuke dingen doen samen.

Liefste Anne-Marie, wat hebben we samen al een avonturen beleefd. Keuzes werden vaak bepaald omdat ik iets in mijn hoofd had. Toch hebben wij overal ons huisje kunnen bouwen. Jij bent mijn rustpunt en mijn aarde. Dank je voor je steun en liefde.

Ik heb gezegd.