Evaluatie van Part-of-Speech taggers voor het Corpus Gesproken Nederlands

Jakub Zavrel
ILK/Taal & Informatica
Katholieke Universiteit Brabant
zavrel@kub.nl

Walter Daelemans
ILK/Taal & Informatica en
Centrum voor Nederlandse Taal en Spraak
Universitaire Instelling Antwerpen
walter@kub.nl

1 juli 1999

Dit document is het eindverslag van de evaluatie van bestaande POS taggers voor het Nederlands voor de subwerkgroep Tagging.

1 Inleiding

Een van de annotatieniveaus van het Corpus Gesproken Nederlands (CGN) gaat bestaan uit gedisambigueerde morfo-syntactische woordcategorieën (tags). Om het taggen van de grote hoeveelheid materiaal in het CGN mogelijk te maken, moet gebruik worden gemaakt van een automatische tagger. De taak van deze tagger bestaat uit het leveren van lexicale informatie (de mogelijke categorieën van een woord) en het, waar nodig, disambigueren op basis van de context. Aangezien het doel van het project de productie van een corpus is, is de belangrijkste eis aan de tagger dat deze het annotatie- en correctieproces zo accuraat en efficient mogelijk maakt. De tagger moet zo weinig mogelijk fouten maken, en het moet de menselijke annotatoren eenvoudig gemaakt worden de gemaakte fouten te corrigeren. Voor een uitgebreidere beschrijving van de taak en de randvoorwaarden verwijs ik naar de meest recente versie van het stuk *"CGN-tagset"*[Van Eynde 1999], dd. 10/05/99.

Er is op het moment een heel scala aan tagger-technologie beschikbaar. Het is op dit punt belangrijk een onderscheid te maken tussen de *tagger* (dwz. een stuk software dat taalkennis bevat en ruwe tekst van tags kan voorzien), en een *tagger-generator* (dwz. een stuk software dat getraind wordt op een getagd corpus en een tagger oplevert). Er bestaan een aantal kant en klare taggers voor het Nederlands (nader beschreven in Sectie 3). De meeste van deze taggers zijn statistisch of lerend van aard, en zijn dus gemaakt door een tagger-generator, die in de meeste gevallen ook beschikbaar is. We noemen deze dus ook *trainbare taggers*. Daarnaast zijn er tagger-generatoren die nog nooit op het Nederlands zijn toegepast, maar die ook in aanmerking komen.

De meeste state-of-the-art (trainbare) taggers halen een accuraatheid van tussen de 90 en 97% procent (afhankelijk van tagset en hoeveelheid trainingsmateriaal, meestal gemeten over alle teksttokens, dus inclusief niet ambigue woorden en punctuatie)[1]. Deze percentages

[1]Voor de regelgebaseerde ENGCG tagger van Lingsoft wordt zelfs een accuraatheid van boven de 99% geclaimd. Deze is echter niet beschikbaar voor het Nederlands. De ontwikkeltijd van een Constraint Grammar

worden echter vrijwel altijd gemeten over een achtergehouden stuk van het trainingcorpus. Het is een publiek geheim dat wanneer taggers toegepast worden op een anderssoortig domein of corpus, de accuraatheid aanzienlijk minder wordt (er zijn anecdotische schattingen bekend van 60-90%). Met name het feit dat de meeste bestaande taggers getraind zijn op tekst, terwijl het CGN uit getranscribeerde spraak bestaat kan een punt van zorgen zijn (cf. [Nivre *et al.* 1996]).

Het voornaamste probleem bij het selecteren van een tagger voor het CGN is echter op het moment een kip-ei situatie. De bestaande taggers werken geen van allen met de beoogde CGN tagset. Bovendien zijn de bestaande tagsets voor een deel veel minder uitgebreid dan die van het CGN. Als er al een flink stuk corpus beschikbaar zou zijn, geannoteerd in de CGN tagset, zou het zonder meer de beste weg zijn om direct een nieuwe tagger (of meerdere, zie Sectie 4) hierop te trainen—dit zou immers het probleem oplossen van de ondergespecificeerde m-naar-n mapping (waarbij m over het algemeen veel kleiner is dan n) naar de CGN tagset. Om echter zonder tagger een stuk corpus te annoteren van een hiervoor bruikbare grootte (50-200k woorden), is een redelijk inspannende klus.

De selectie van een tagger dient dus op het moment twee doelen/fases:

1. Om een initieel deel van het CGN corpus te bootstrappen vanuit een bestaande tagset, en
2. De annotatoren in latere fases behulpzaam te zijn met de annotatie in de CGN-eigen tagset.

Voor het eerste doel of fase moet er een reeds beschikbare tagger voor het Nederlands geselecteerd worden. Aangezien er vanaf een bepaald moment genoeg materiaal zal zijn om een CGN-eigen tagger te trainen, en zo het tweede doel beter te dienen, dient ook te worden gekeken naar tagger-generatoren.

De rest van dit document is als volgt opgebouwd. Sectie 2 geeft een overzicht van de evaluatiecriteria. Sectie 3 geeft een overzicht van de beschikbare taggers en tagger-generatoren, en is gedeeltelijk overgenomen uit een eerder stuk van Ineke Schuurman ("Overzicht Taggers", Sectie 3 behorende bij "CGN-tagset" [Van Eynde 1999]). Deze sectie is tevens aangevuld met verdere informatie die van de taggerleveranciers is verkregen[2] . Sectie 4 bespreekt de mogelijkheden van een synergie tussen verschillende taggers in beide fases. Sectie 5 geeft de experimentele opzet weer en beschrijft de quantitatieve en qualitatieve resultaten van de empirische evaluatie. Sectie 6 bespreekt de consequenties voor de CGN tagset. Sectie 7, tenslotte, bespreekt deze resultaten met het oog op de objectieven van het CGN, trekt op basis van de bevindingen conclusies. In de Appendices zijn de tagsets van verschillende beschikbare taggers opgenomen.

2 Evaluatie-criteria

Er zijn weinig vaststaande en algemeen geldende criteria voor tagger-evaluatie te vinden. Vaak wordt er alleen gekeken naar accuraatheids percentages (bv. in de Franse GRACE

voor een nieuwe taal is aanzienlijk, en derhalve is de ontwikkeling hiervan binnen het CGN project waarchijnlijk niet haalbaar.

[2]Met dank aan: Theo Vosse, Isa Maks, Jean-Pierre Chanod, Agnes Sandor, Bob Boelhouwer, Theo van den Heuvel, Thorsten Brants en Hans van Halteren

evaluatie [GRACE]). Verder is het voornamelijk belangrijk een tagger te vinden die goed aansluit bij de taak, het domein, en de gewenste fijnkorreligeheid van analyse. Zoals in de inleiding al is aangegeven zijn er ons inziens twee doelen/fases in het project: het bootstrappen van het corpus en het helpen van de annotatoren in de productiefase. In de bootstrapfase hebben we te maken met bestaande taggers, die allen werken met een andere tagset dan de CGN tagset, en moeten we deze deze toepassen op een testcorpus, bestaande uit initiële CGN transcripties, om te kijken hoe goed ze het doen. Op het moment zijn de eerste referentie-transcripties van het CGN al beschikbaar (ongeveer 10k woorden) en deze zijn voor een deel met de hand geannoteerd in de CGN-tagset [Zavrel 1999]. Deze evaluatie is echter niet triviaal, omdat het moeijlijk is verschillende taggers te evalueren als zij verschillende tagsets gebruiken. Er zijn dus twee componenten van accuraatheid:

- **accuraatheid in termen van de eigen tagset.** Deze kan gemeten worden door puur te kijken of de aan het testmateriaal toegekende tags niet foutief zijn. Dit wordt lastig gemaakt door het feit dat de tagsets en tagging criteria vaak niet of nauwelijks gedocumenteerd zijn.
- **(geschatte) accuraatheid na mapping op de CGN tagset.** Een eenvoudige tagset kan immers 100% correct zijn toegekend, maar laat nog veel van de in de CGN tagset aanwezige informatie ondergespecificeerd. Dit kan gezien worden als een residu van ambiguiteit. De complexiteit en automatiseerbaarheid van de mapping spelen uiteraard ook een belangrijke rol bij de selectie.

In de latere fase van het annoteren, dwz. als er al genoeg materiaal beschikbaar is om een CGN-eigen tagger te trainen spelen weer andere criteria een rol. Natuurlijk is er het verschil in accuraatheid tussen verschillende tagging technieken, zelfs wanneer deze getraind worden op exact hetzelfde materiaal (zie [Van Halteren *et al.* 1998]). Maar men kan zich afvragen of kleine verschillen in accuraatheid de belangrijkste factoren zijn bij het efficienter maken van het annoteren van grote hoeveelheden getranscribeerd materiaal. Belangrijke(re) criteria in dit verband zijn bijvoorbeeld de mogelijkheid tot adaptatie van de tagger (zodat deze steeds beter aangepast is aan het type materiaal waarmee gewerkt wordt), en de mogelijkheid de tagger op een effectieve manier toe te passen binnen een geschikte annoteer-tool. Neem bijvoorbeeld een tagger die altijd een tag teruggeeft met een accuratesse van 98% maar die geen waarschijnlijkheden aan zijn keuzes meegeeft. De correctie van deze uitvoer zal waarschijnlijk trager (en dus duurder) zijn dan die van een tagger die een procent minder accuraat is, maar onzekere beslissingen duidelijk aangeeft en alternatieven suggereert.

Om een indruk te krijgen van de geschiktheid van de verschillende taggers mbt. dit soort randvoorwaarden, zijn de volgende criteria/vragen opgesteld die aan de tagger leveranciers zijn voorgelegd. Daarnaast is aan alle leveranciers ook de vraag gesteld wat eventuele voorwaarden, prijzen en beperkingen zouden zijn om een licentie te krijgen voor het gebruik van de betreffende tagger(generator) in het CGN project. Ondertussen hebben alle taggerleveranciers deze vragen beantwoord. De antwoorden zijn verwerkt in het overzicht van bestaande taggers in Sectie 3

Vragen/Criteria: (*motivatie*)

- Welke tagset(s) gebruikt de tagger? Is er documentatie beschikbaar voor de tagset (en de tagger)? *Om in te schatten hoe complex de mapping naar de CGN tagset zal zijn.*

- Wat voor soort domein/corpus is gebruikt om de tagger te trainen? *Om in te schatten of er veel fouten zullen optreden door domein-mismatch.*
- Doet de tagger ook aan tokenisatie en lemmatisatie? *Het eerste is niet perse belangrijk in de context van het CGN, tenzij de tagger afhankelijk is van een tokenisatie die sterk afwijkt van die van de CGN transcriptie. De lemmatisatie is belangrijk in verband met de lexicon-koppeling.*
- Geeft de tagger soms meer dan een tag terug? Zo ja, wordt er dan een mate van zekerheid aangegeven? Geeft de tagger soms geen enkele tag terug? Dit in verband met de integratie in een annotatie-tool.
- Hoe worden onbekende woorden door de tagger behandeld?
- Is het mogelijk om de tagger opnieuw te trainen op nieuw materiaal (andere tagsets)? *(dwz. is de tagger geassocieerd met een tagger-generator?)*
- Is het mogelijk om incrementeel informatie toe te voegen aan het lexicon en aan de context constraints? *Dit in verband met de incrementele adaptatie naarmate de hoeveelheid CGN materiaal toeneemt.*
- Heeft de tagger mogelijkheden om idiomen en andere multi-word uitdrukkingen speciaal te behandelen? *Indien er speciale behandeling is voor bijvoorbeeld vaste uitdrukkingen of klasses van expressis (namen, data, etc.), dan is het mogelijk om tagfouten hierop systematisch te vermijden.*
- Hoe snel is de tagger ongeveer tijdens het trainen en testen? (woorden/sec). *Het moet niet buitengewoon traag gaan.*
- Onder welke operating systemen draait de tagger?
- Is het mogelijk om de tagger-software aan te passen aan specifieke I/O vereisten (beschikbaarheid source code/API)?
- Kan de tagger met SGML mark-up overweg?

3 Beschikbare taggers en tagger-generatoren

In het document *"Overzicht taggers"* van Ineke Schuurman, dd. 5/11/98, wordt een overzicht gegeven van beschikbare taggers voor het Nederlands. Dit overzicht wordt hier met enige wijzigingen overgenomen, aangevuld, en geïntegreerd met de informatie die taggerleveranciers gaven naar aanleiding van bovenstaande vragenlijst.

3.1 D-Tale tagger

.

De Dutch TAgger-LEmmatizer voor het Nederlands is een programma dat werd ontwikkeld door de afdeling lexicologie aan de VU, voor Van Dale. Het is een regelgebaseerde tagger/lemmatizer met een groot woordvormenlexicon (140.000 woordvormen) en een vaste eigen tagset (zie Appendix A.3 voor een beschrijving). Deze tagger is niet opnieuw trainbaar,

aangezien de disambiguatieregels met de hand zijn opgesteld (In een Constraint Grammar formalisme à la [Karlsson *et al.* 1995]), maar is eventueel wel handmatig aan te passen. De disambiguatieregels elimineren tags uit kandidaten die door het lexicon zijn voorgesteld. Bij niet opgeloste ambiguïteit wordt meer dan één tag teruggegeven, zonder mate van zekerheid. In enkele gevallen worden alle kandidaten geëlimineerd. Het lexicon is uitbreidbaar, en kan ook multi-word units bevatten. D-Tale is ontwikkeld en getest aan de hand van algemene teksten: kranten, stukken wetenschappelijk proza, en scripts van soap-series.

Het systeem is geschreven in C en draait onder UNIX, en is dus mogelijk aan te passen aan specifieke eisen van een annoteeromgeving. De tagger is, zo werd door een van de ontwikkelaars geantwoord, "niet al te snel". Het systeem is beschikbaar voor gebruik door het CGN.

3.2 Duchtale/PAROLE tagger

Het INL beschikt over een Nederlandse tagger Dutchtale, gebaseerd op een lexicon, morfologische analyse, en een hybride van een regelgebaseerde en statistische disambiguatiecomponent. Op het moment wordt er gewerkt aan een modernere opvolger op basis van de PAROLE tagset en HMM-technnieken. Het INL heeft laten weten de verouderde Dutchtale tagger niet te willen laten deelnemen aan de evaluatie. De opvolger, de nieuwe PAROLE tagger, was echter niet op tijd klaar om deel te nemen aan de evaluatie.

3.3 Xerox tagger

Xerox Research Center Europe heeft trigram-gebaseerde taggers ontwikkeld voor een groot aantal Europese talen, waaronder ook het Nederlands. De Nederlandse tagger is beschikbaar onder een onderzoekslicentie voor een eenmalig bedrag van US $ 2000 voor één jaar, gratis verlengbaar. Updates en nieuwe versies zullen opnieuw betaald moeten worden. Deze tagger is gebaseerd op deterministische finite state automaten, vergelijkbaar met de tagger beschreven in [Chanod & Tapanainen 1995], en is in principe hertrainbaar. De tool die hiervoor en voor incrementele aanpassing aan het lexicon benodigd is, valt echter buiten de genoemde onderzoekslicentie, tenzij er een samenwerkingsovereenkomst met Xerox wordt afgesloten. De tagger gebruikt een door Xerox ontwikkelde eigen tagset (zie Appendix A.4), heeft een eigen tokenisatiemodule en doet tevens lemmatisering. Het is mogelijk om multi-word units in het lexicon op te nemen. De tagger is deterministisch en geeft dus telkens slechts een tag terug, zonder een mate van zekerheid. Zowel bekende als onbekende woorden worden morfologisch geanalyseerd om de mogelijke tags te bepalen.

De tagger is geïmplemteerd als onderdeel van het XeLDA client/server platform dat een goed gedocumenteerde API heeft. Het systeem draait zowel onder UNIX als onder Windows. De snelheid is in de orde van duizenden woorden per seconde voor tagging en enkele seconden voor hertrainen.

3.4 Brill tagger

Deze tagger (generator), gemaakt door Eric Brill van John Hopkins University, is gratis te downloaden van het internet (C en Perl source code). Zowel in Groningen als in Tilburg is deze tagger reeds toegepast (getraind) op WOTAN-I materiaal. In tegenstelling tot veel van de andere technieken, genereert deze tagger-generator regels die met enige inspanning door mensen te begrijpen zijn. De tagger geeft echter steeds maar een tag terug, zonder mate van

zekerheid. Het trainen van de tagger op nieuw materiaal is kubiek in de grootte van de tagset, en voor een tagset als die van het CGN bleek Brill's implementatie zo langzaam te zijn dat deze niet mee is genomen in de empirische evaluatie.

3.5 KEPER tagger

De KEPER tagger is een bigram-tagger gecombineerd met morfologische analyse, ontwikkeld door Polderland BV. Deze tagger is beschikbaar voor CGN voor een vaste prijs van Hfl. 18000,- en aanpassingen tegen uurtarief. KEPER werkt met een eigen tagset (zie Appendix A.2), die ontwikkeld is met het oog op Information Retrieval toepassingen, en is getraind op een bestand van formeel geredigeerd Nederlands. Het systeem is hertrainbaar op een nieuw corpus met een andere tagset, maar dit zou door Polderland uitgevoerd moeten worden. De training van de statistische parameters kan incremeteel gebeuren, maar het tagging-lexicon is vast geïntegreerd in het systeem. Wel kan de gebruiker met een aparte lexiconfaciliteit een eigen woordenlijst toevoegen, die het systeemlexicon kunnen 'overrulen'. Het trainen van de tagger op een nieuw corpus gebeurt onder supervisie van een menselijke operator. Over de trainingssnelheid meldt Polderland: *"Bij begin van de training ligt het tempo rond de 5 woorden per seconde. Het loopt snel op. Later neemt de snelheid geleidelijk minder toe en convergeert tot het eindtempo dat een veelvoud van het begintempo is. Trainingsmateriaal kan hergebruikt worden, waarbij het tempo bijna zo hoog is als voor de gewone productiesnelheid. Aangezien we KEPER nog niet hebben getraind op andere soorten tekst hebben we nog weinig ervaring met het trainingstempo. De trainingsfaciliteit is eenvoudig te bedienen."*

De KEPER-tagger doet zowel tokenisatie, lemmatisatie en tagging. De tagger produceert slechts een enkele tag per woord, tenzij disambiguering wordt uitgeschakeld, dan geeft hij alle mogelijke tags terug.

Het systeem draait onder Windows en diverse UNIX varianten en is op aanvraag ook voor andere platforms geschikt te maken. De taggingsnelheid is 45 minuten voor 1 Mbyte tekst op een SUN SPARCstation uit 1992. Het outputformaat is aan te passen met behulp van een style-file. Keper maakt optioneel gebruik van een gecustomiseerde preprocessor. De standaard input is gewone ASCII.

3.6 CORRie tagger

Deze tagger generator, ontwikkeld door Theo Vosse in Leiden, is een standaard trigram HMM tagger, plus "een poging om structuur te herkennen aan de hand van functiewoorden." Onbekende woorden worden gegokt op basis van het einde van het woord. De tagger is gebaseerd op een kleine tagset van hoofdcategorieën (zie Appendix A.1) en is getraind op één miljoen woorden handmatig door INL geannoteerde krantenteksten. De software is in principe beschikbaar voor CGN, al zou er nog overlegd moeten worden over de precieze condities, aangezien de trainingscorpora eigendom zijn van INL. De tagger kan eenvoudig getraind worden op een nieuw corpus met een andere tagset. De tagger kan de N-beste tags per woord teruggeven, voorzien van hun waarschijnlijkheid.

De software is portable naar allerlei platforms en het is in principe mogelijk deze aan te passen aan specifieke vereisten van de annoteeromgeving.

3.7 WOTAN tagger

Deze tagger (generator), gebaseerd op Hidden Markov Modellen (HMM) en een memory-based module voor onbekende woorden, is ontwikkeld bij de afdeling Taal en Spraak van de KUN. Op het moment is deze tagger getraind op het Eindhoven corpus geannoteerd met de WOTAN-I tagset. Een nieuwe versie, gebaseerd op de uitgebreidere WOTAN-II tagset is in de maak, maar was niet op tijd klaar om deel te nemen aan de evaluatie. Aangezien ondertussen is gebleken dat de HMM-implementatie van de TnT tagger (zie hieronder) superieur is, is de oudere versie van deze tagger terugetrokken uit de evaluatie.

3.8 Memory-Based tagger

De MBT tagger (generator), ontwikkeld door de ILK groep aan de KUB en UIA, werkt op basis van Memory-Based Learning, analogisch redeneren met in het geheugen opgeslagen voorbeelden. Deze tagger is eenvoudig hertrainbaar op een nieuw corpus, en is momenteel beschikbaar met de WOTAN-I tagset (getraind op 600 duizend woorden Eindhoven Corpus) en WOTAN-II (getraind op 150 duizen woorden) (zie Appendix A.5 en [Van Halteren 1999] voor meer informatie over de WOTAN tagsets). Het lexicon is incrementeel uitbreidbaar. De tagger tokeniseert niet, en lemmatiseert niet, maar er is een tokenisatie preprocessor en een aparte Memory-Based morfologische analyse (MBMA) module beschikbaar (zie http://ilk.kub.nl/ voor een demo) die extrapoleert vanuit CELEX. De tagger kan meerdere tags per woord teruggeven, vergezeld van een zekerheidsmaat. Onbekende woorden worden gegokt op basis van een aantal vorm-features (prefix, hoofdletter?,suffix, getallen?, hyphen?).

Het systeem is draait onder verschillende UNIX varianten en is gemakkelijk te porten naar andere platforms (C++ code en Perl scripts). Het trainen duurt enkele minuten, en de snelheid bij het taggen is meer dan twintigduizend woorden per seconde op een Pentium II PC. Er wordt gewerkt aan een versie die SGML invoer en uitvoer aankan.

3.9 MXPOST (Maximum Entropy tagger)

Deze tagger (generator), gemaakt door Adwait Ratnaparkhi van UPenn, is in Tilburg toegepast (getraind) op WOTAN-I en WOTAN-II materiaal. De tagger is eenvoudig trainbaar op nieuwe corpora, maar heeft geen expliciet lexicon (dus ook niet een incrementeel uitbreidbaar lexicon). Onbekende woorden worden gegokt op basis van een aantal vorm-features. In een vergelijk van vier bekende taggertechnieken (van Halteren et al., 1998) bleek deze tagger steeds significant accurater dan de andere (Brill, HMM, MBT). Recentere experimenten laten echter zien dat TnT (zie hieronder) vaak net iets accurater is.

De software is voor onderzoeksdoeleinden gratis te downloaden van het internet (voorgecompileerde Java bytecode, dus platform onafhankelijk). Het trainen van de tagger duurt voor een groot corpus enkele dagen. Taggen is ongeveer een seconde per zin van 30 woorden op een Solaris Pentium II PC.

3.10 TnT (Trigrams 'n Tags)

TnT is een trigram HMM tagger, ontwikkeld door Thorsten Brants van de Universiteit van Saarbrücken. De software is een onderdeel van het Annotate platform voor corpus-annotatie, en is gratis beschikbaar voor onderzoeksdoeleinden. De tagger maakt gebruik van lineaire interpolatie voor smoothing van kansen en van "successive abstraction" [Samuelsson 1996] op

suffixen van woorden om onbekende woorden te gokken. Het trainen van de tagger is een kwestie van enkele seconde en taggen gaat met enkele duizenden woorden per seconde. Deze tagger is in Tilburg getraind op WOTAN geannoteerd materiaal en hij bleek even accuraat als, of zelfs superieur aan de MXPOST tagger.

4 De synergie van tagger-combinatie

Zoals reeds eerder door Hans van Halteren en Walter Daelemans naar voren is gebracht, kan het een groot voordeel zijn om over meerdere taggers te beschikken en de resultaten hiervan te combineren. De reden hiervoor is dat de fouten van taggers met verschillende architecturen in een bepaalde mate ongecorreleerd zijn, waardoor de taggers elkaar als het ware kunnen corrigeren. Er vallen twee toepassinggebieden voor dit concept aan te wijzen, analoog aan de twee fases van de annotatieinspanning. [Van Halteren *et al.* 1998, Brill & Wu 1998] hebben laten zien dat een gecombineerde tagger een reductie van het aantal fouten tussen de 10 en 20 % laat zien[3]. Verder hebben [Màrquez *et al.* 1998] laten zien dat juist de punten waar de verschillende taggers niet overeen komen interessante cases zijn, die de aandacht van de menselijke annotator waard zijn. De mogelijkheid in de eerste (bootstrap) fase is echter ook interessant. Deze berust op het idee dat het probleem met de meeste, nu beschikbare, taggers is dat hun tagset (bv.Xerox, WOTAN-I) minder verfijnde onderscheiden maakt dan de beoogde CGN tagset. In het geval dat een zelfde maat van fijnkorreligheid wel aanwezig is (bv. WOTAN-II), dan worden vele onderscheiden langs een andere lijn getrokken. Wanneer we een klein initieel deel van het corpus laten annoteren door de verschillende taggers (in hun eigen tagset) dan valt uit te maken of de doorsnede van de tagsets (en de fouten) wellicht veel makkelijker af te beelden is op de vrij gedetailleerde CGN tagset. Een experiment langs deze lijnen is uitgevoerd en wordt beschreven in Sectie 5.6.

5 Experimenten

Naar aanleiding van beschikbaarheid, haalbaarheid, enige voorselectie zijn de volgende taggers opgenomen in de experimenten:

- D-Tale: testtagging uitgevoerd door Isa Maks van de VU.
- KEPER: testtagging uitgevoerd door Polderland BV.
- Xerox: testtagging uitgevoerd door Agnes Sandor van XRCE. Vanwege reglemetaire beperkingen (max. samplegrootte 3000 tokens) en gedeeltelijke mismatch van de door Xerox getagde sample met de handmatig gannoteerde selectie kon Xerox slechts op een kleine tweeduizend tokens (waarvan 1666 echte woorden) geëvalueerd worden.
- CORRie: testtagging uitgevoerd door Theo Vosse van de RUL. Alle leestekens waren hierbij weggevallen, en zijn later weer terugezet (tellen echter toch niet mee voor de evaluatie).

[3]Helaas zij we door tijdgebrek, en omdat we het idee hadden dat toch vooral zou gaan om de verschillen in tagset (en WOTAN taggers zijn al ruim vertegenwoordigd) niet toegekomen aan het meenemen van de combi-tagger in de evaluatie. Het valt evenwel te verwachten dat de resultaten hiervan iets beter zouden zijn.

- WOTAN-I taggers: MBT, MXPOST en TnT, getraind op het WOTAN-I geannoteerde deel van het Eindhoven corpus (>600k woorden). Testtagging door Jakub Zavrel van de KUB. De MBT tagger is door Antal van den Bosch verrijkt met uitvoer van de MBMA lemmatiseerder.
- WOTAN-II taggers: MBT, MXPOST en TnT, getraind op het WOTAN-II geannoteerde deel van het Eindhoven corpus (±150k woorden). Testtagging door Jakub Zavrel van de KUB.

Om de accuraatheid van de verschillende taggers te meten zijn een aantal experimenten uitegevoerd op een kleine, met de hand geannoteerde sample van CGN materiaal. In Sectie 5.1 wordt de gebruikte testdata beschreven. Vervolgens wordt in Sectie 5.2 van de taggers die tevens lemmatiseren de accuraatheid op dit onderdeel gegegeven. In Sectie 5.3 wordt de accuraatheid van de taggers in termen van hun eigen tagset gemeten. In Sectie 5.4 wordt op basis van de uitvoer van de taggers de CGN tag voorspeld, om een schatting te krijgen van de complexiteit van de mapping naar de CGN tagset. Ondanks de zeer kleine hoeveelheid reeds met CGN tags geannoteerde data is het toch interessant om de accuraatheid van tagger-generatoren op dit materiaal te zien. Dit wordt gemeten in Sectie 5.5. Tot slot is er een experiment uitegevoerd met een machine-learning methode voor de combinatie van de uitvoer van alle gebruikte taggers (Sectie 5.6).

5.1 Data

In de experimenten is een gedeelte van de eerste referentie-transcripties gebruikt die voor het CGN gemaakt zijn. Er was een totaal van iets meer dan tienduizend woorden beschikbaar, in verschillende genres van gesproken taal. Deze zijn automatisch in zinnen gesplitst en getokeniseerd[4]. Vanwege tijdsbeperkingen is besloten om hiervan slechts een deel door alle drie de annotatoren te laten verwerken. Er zijn een kleine drieduizend tokens drievoudig geannoteerd: de zinnen 1 t/m 130 (grofweg de eerste tweeduizend tokens van het Nederlandse deel), en de zinnen 592 t/m 677 (ongeveer de eerste duizend tokens van het Vlaamse deel) van de transcripties. Dit zijn in totaal 2985 tokens, waarvan 2388 woorden (na weglating van leestekens en niet te annoteren markup-tekens. De drie annotatoren waren vrijwilligers uit de subwerkgroep POS tagging. Er is gewerkt met twee verschillende annoteeromgevingen: twee annotatoren werkten met `cgntsel`, geschreven door Hans van Halteren. Dit programma is specifiek toegesneden op het taggen en lemmatiseren van tekst. De derde annotator werkte met Annotate [Plähn 1998], een tool met grafische interface die tevens gebruikt kan worden voor annotatie van syntactische structuur. Annotate kan echter op het moment niet gebruikt worden om lemma-informatie toe te voegen. Alle annotatoren werkten met de tagset zoals gedefinieerd in [Van Eynde 1999] (versie van 10 mei 1999). De complete geannoteerde data inclusief de drie versies van de annotatie is beschikbaar op URL `http://ilk.kub.nl/~zavrel/agreement.html`. Een uitgebreidere beschrijving van het annoteerexperiment is te vinden in [Zavrel 1999].

De handmatige lemmatisering is slechts door twee annotatoren uitegevoerd, waarbij een overeenstemming van 97.7% werd bereikt. Hoewel de drie annotatoren in vrij veel gevallen verschillende tags toekenden (slechts 76.3% totale overeenstemming, 93.4% op hoofcategorie), waren zij het in slechts 29 (1.2%) gevallen alle drie niet met elkaar eens. Uitgaande hiervan

[4]Met dank aan Sabine Buchholz voor het beschikbaar stellen van de tokenizer.

hebben wij een tagging geconstueerd die uitgaat van de meerderheid van de drie annotatoren, met een extra handmatige correctie voor de gevallen van absoluut verschil. Deze data dient als testset voor de experimenten hieronder.

5.2 Lemmatisering

Slechts vier van de deelnemende taggers (MBT,D-Tale,Xerox en KEPER) doen aan lemmatisering. In Tabel 1 zien we de resulaten van de vergelijking met de menselijke lemmatisering.

	number of errors (% wrong)			
tag	MBT	D-Tale	Xerox	KEPER
WW	347 (92.2)	42 (11.1)	15 (3.9)	271 (72.0)
N	24 (6.6)	31 (8.5)	47 (12.9)	33 (9.1)
ADJ	40 (19.9)	18 (9.0)	12 (6.0)	36 (17.9)
VNW	5 (1.3)	9 (2.3)	17 (4.3)	5 (1.3)
BW	–	2 (0.8)	5 (2.0)	–
VZ	–	–	–	3 (1.2)
VG	–	–	–	–
TSW	3 (1.7)	6 (3.3)	1 (0.5)	24 (13.3)
TW	5 (20.0)	9 (36.0)	3 (12.0)	3 (12.0)
LID	–	–	8 (4.4)	–
total	435 (18.2)	128 (5.3)	113 (6.7)	386 (16.1)

Tabel 1: Aantallen fouten van de lemmatisers, uitgesplitst per POS tag. De getallen tussen haakjes in de kolommen geven aan welk percentage van die POS tag verkeerd werd gelemmatiseerd.

Alleen D-Tale en Xerox vallen binnen redelijke normen van accuraatheid met respectievelijk 5.3 en 6.7% fout. Wanneer we echter de fouten opsplitsen naar POS tag dan zien we dat het verschil voornamelijk ligt in de categorie werkwoorden. Een nadere inspectie van de uitvoer leert ons dat MBT en KEPER werkwoorden reduceren naar hun stam, terwijl de overige twee lemmatisers naar de infinitiefvorm reduceren, hetgeen de menselijke annotatoren ook deden. Dit is een min of meer willekeurige keuze, oa. ingegeven door het feit dat MBT en KEPER beide CELEX volgen. Verder telt zwaar mee dat MBT "is" als lemma aanhoudt, terwijl alle overigen dit naar "zijn" lemmatiseren. Wanneer we de werkwoorden niet meerekenen ontlopen de methodes elkaar niet veel met resterende totale foutpercentages van 3.6%, 3.6%, 5.8% en 4.8% respectievelijk voor MBT, D-Tale, Xerox en KEPER. Dit betekent overigens niet dat er geen daadwerkelijke fouten in de lemmatisering van werkwoorden voorkomen (bv. "gedacht – gedenken","leek – leek (N)"). Dat evaluatie van lemmatisering toch enigszins arbitrair is wanneer niet duidelijk is uit welke lijst de lemma's geselecteerd dienen te worden, kunnen we nog duidelijker maken aan de hand van het woord "'t", dat door mens en drie lemmatisers zo werd gelaten, maar door Xerox correct naar "het" werd gelemmatiseerd. Hetgeen 8 fouten oplevert in bovenstaande score.

5.3 Taggers op hun eigen criteria gemeten

Hoewel alle taggers een andere tagset gebruiken dan de beoogde CGN tagset, en een goede accuraatheid geen eenvoudige vertaling naar de doeltagging garandeert, zijn we toch geïnteresseerd in de mate waarin zij hun eigen tagset juist toepassen. Om dit echter te meten terwijl we geen correcte referentietagging van het testmateriaal in die tagset voorhanden hebben, moeten we uitgaan van de "correcte" CGN tagging en op basis hiervan evalueren. Dit is gedaan door de taggeruitvoer naast de CGN-tagging te leggen en slechts die paren van tags af te keuren die duidelijk met elkaar in tegenspraak zijn. Wanneer een tagger dus "finiet werkwoord" als tag geeft, en een aparte tag voor "infinitief" heeft zal dit fout gerekend worden als CGN "infinitief" aangeeft. Wanneer er echter bv. geen eigen tag voor "hulpwerkwoord" is dan zal dit niet aangerekend worden. Een tagger die minder gedetaileerde informatie teruggeeft heeft dus ook minder kans om fouten te maken. Bij verschillen in opvatting over de grens van categorieën is de CGN grens als ijkpunt gekozen. Wanneer een tagger bijvoorbeeld zegt dat een woord een "naamwoord" is (zoals bv. KEPER die alle getallen als naamwoorden tagt) terwijl CGN zegt dat dit een "telwoord" is dan is dit fout, ook al is de tagger daarin consistent. Omdat dit subjectieve oordelen kan behelzen, is er uiterst soepel te werk gegaan. Alles wat niet duidelijk fout is, is goed. Wanneer een tagger (D-Tale was hier het enige voorbeeld van) ambiguiteit overlaat door n tags terug te geven dan wordt dit $1/n$ goed (of fout) gerekend.

Tag	Freq	Tagger-accuraatheid									
						WOTAN-I			WOTAN-II		
		D-Tale	KEPER	XEROX	CORRie	MBT	MX	TnT	MBT	MX	TnT
N	364	92.6	85.2	96.4	97.8	93.4	95.3	97.8	91.5	89.6	90.1
ADJ	201	83.1	79.6	81.5	89.1	81.6	81.6	81.1	92.0	84.1	90.5
WW	376	88.4	72.3	74.0	91.8	84.8	87.2	85.4	73.7	68.6	72.3
BW	246	88.4	73.2	87.4	87.0	91.9	93.5	94.7	88.6	94.3	94.7
TW	25	72.0	0.0	73.3	68.0	76.0	88.0	80.0	84.0	80.0	88.0
TSW	180	30.0	27.2	16.0	23.9	86.7	55.6	91.1	26.7	25.0	24.4
VNW	399	79.2	62.9	69.3	90.0	77.9	82.5	82.0	85.2	85.7	89.5
LID	182	93.7	95.6	95.7	96.2	99.5	96.7	100	97.3	100	100
VZ	244	82.2	87.3	86.1	89.8	89.3	90.2	89.3	89.8	89.3	90.6
VG	160	89.4	87.5	84.0	95.0	95.0	92.5	94.4	94.4	94.4	94.4
TOT	2388	82.4	73.7	78.8	86.7	87.8	86.9	89.9	82.9	81.8	83.9

Tabel 2: De accuraatheid van taggers met betrekking tot hun eigen tagset, gemeten op het CGN getagde materiaal aan de hand van een "soepele" mapping naar de CGN tagset. NB: de XEROX tagger is op slechts 1666 woorden geëvalueerd.

De resultaten zijn weergegeven in Tabel 2. We zien dat de taggers met een redelijk grove tagset (niet-WOTAN) niet persé veel beter scoren dan de gedetailleerde WOTAN taggers. Een uitzondering is CORRie, met 86.7% maar deze geeft slechts de 11 hoofdcategorieën terug. De beste tagger is hier de WOTAN-I TnT tagger met 89.9% correct. Getest op een testset uit het Eindhoven corpus scoorde deze tagger 92.1%; het verschil is niet dramatisch groot. De scores van de WOTAN-II taggers zijn beduidend lager, waarschijnlijk omdat deze op veel minder data getraind zijn.

5.4 De mapping naar de CGN tagset

De scores in de vorige sectie zijn op zich al interessant, maar laten zich lastig vertalen naar een tagging-accuraatheid in termen van de CGN tagset. In werkelijkheid zou een tag van tagger X vertaald kunnen worden naar een aantal mogelijke CGN tags, enerzijds een aantal meer specifieke tags, anderzijds een aantal verschillende interpretaties. Er is dus een mate van onzekerheid over, nadat we de tag van X ontvangen hebben. Dit laat zich uitdrukken in de Information Gain (IG) van de tagger. De Information Gain is het verschil in Entropie (onzekerheid) van de annotator over de tag, voordat en nadat deze de tag van X heeft gezien. IG wordt gemeten met behulp van vergelijking 1.

$$IG(X) = H(CGN) - \sum_{x \in Tags(X)} P(x) \times H(CGN|x) \tag{1}$$

Waar $H(CGN)$ de entropie is van de CGN tagset, $P(x)$ de kans dat de tagger tag x kiest en $H(CGN|x)$ de onzekerheid van in de CGN keuze gegeven tag x. Omdat een tagset met meer tags automatisch meer IG heeft, moeten we wanneer we taggers met verschillende tagsets vergelijken normaliseren voor het aantal waarden. De genormaliseerde waarde heet Gain Ratio (GR) [Quinlan 1993].

$$GR(X) = \frac{IG(X)}{si(X)} \tag{2}$$

en de normaliserende factor, splitinfo(x) is de entropie van de tagset van X:

$$si(X) = H(X) = \sum_{x \in Tags(X)} P(x) log P(x) \tag{3}$$

Om de vertaling te kunnen maken naar een accuraatheidspercentage, hebben we ook nog de score berekend wanneer we voor iedere tag van de kandidaat-taggers de meest waarschijnlijke CGN tag kiezen. De resultaten hiervan willen we testen op het CGN materiaal, maar de waarschijnlijkheden worden ook geschat op dat materiaal. Daarom doen we een 10 voudige cross-validatie (10CV). Het testmateriaal wordt in 10 stukken gesplitst en ieder stuk wordt een keer als testset gebruikt, waarbij we de andere 9 stukken gebruiken om te trainen, en we nemen een gemiddelde van de tien stukken. De resultaten zijn te zien in Tabel 3.

					WOTAN-I			WOTAN-II		
	woord	D-Tale	KEPER	CORRie	MBT	MX	TnT	MBT	MX	TnT
IG (bits)	5.21	3.74	2.96	3.00	4.43	4.50	4.60	4.59	4.74	4.79
GR	0.66	0.82	0.80	0.84	0.82	0.84	0.86	0.76	0.77	0.77
accuracy (%)	70.2	50.6	42.8	42.6	71.6	72.6	75.9	72.7	77.2	77.5

Tabel 3: De Information Gain, Gain Ratio, en de accuraatheid van de meest waarschijnlijke CGN tag gegeven de uitvoer van de taggers (10CV).

We zien dat de taggers met een eenvoudige tagset (D-Tale, KEPER, en CORRie)[5] Hun redelijk hoge score op hun eigen tagset (vorige Sectie) niet weten om te zetten naar de CGN tagset. Het te taggen woord zelf is een betere voorspeller van de CGN categorie, hetgeen

[5]Xerox is hier niet meegenomen vanwege de kleinere hoeveelheid data die daarvoor ter beschikking stond.

duidt op een belangrijke rol voor het lexicon in de CGN tagset. De beste voorspeller van de CGN categorie zijn TnT en MXPOST met WOTAN-II tags, gevolgd door de andere WOTAN taggers (MBT-WOTAN-II en de WOTAN-I taggers).

5.5 Trainen op de CGN data

Hoewel we een erg kleine hoeveelheid met CGN categorieën getagd materiaal hebben, leek het toch de moeite waard om eens te proberen de ter beschikking staande tagger-generatoren hierop te trainen. Wellicht weegt het gebruik van de juiste tagset en de overlap van genre en domein op tegen de schaarste aan data. Wederom hebben we een 10CV uitgevoerd, dit keer met MBT, MXPOST, Brill en TnT, waarbij voor iedere "fold" van de 10CV het complete tagger-generatie proces werd doorlopen op 90% van de data. In Tabel 4 zien we het gemiddelde van de tien test-runs.

	MBT	MXPOST	BRILL	TnT
accuraatheid	80.6	69.7	78.2	82.7

Tabel 4: De accuraatheid van de vier beschikbare tagger-generatoren getraind en getest op een kleine sample (2388 woorden) van met de CGN-tagset geannoteerde data (10 voudige cross-validatie).

De verschillen tussen de taggers zijn aanzienlijk. MXPOST is duidelijk de slechtste. Hier zijn twee redenen voor. Ten eerste schat MXPOST de parameters van een vrij complex probabilistisch model, en heeft het dus veel sterker te lijden van "sparse data" dan het eenvoudige trigram model van TnT. Ten tweede gebruikt MXPOST geen features voor de mogelijke categorieën van het te taggen woord, maar alleen contextfeatures. Ook hierdoor komt het model pas goed op gang bij grote hoeveelheden data. Ook hier geeft TnT de andere taggers het nakijken. Het is bovendien verrassend dat TnT op zo een kleine hoveelheid data getraind al ruim uitsteekt (82.7%) boven de resulaten van de mapping in de vorige sectie (beste resultaat van de gemapte tagger was 77.5% (TnT-WOTAN-II).

5.6 Bootstrapping door combinatie

In deze sectie beschrijven we de experimenten met tagger-combinatie, zoals geschetst in Sectie 4. We hebben immers de uitvoer van 10 verschillende taggers (waarvan twee groepen met dezelfde tagset). De patronen van overeenstemming en verschil, fouten en verschillen in granulariteit kunnen met behulp van machine learning methode worden opgepikt. Voorzover het de auteurs bekend is, is dit de eerste keer dat deze methode wordt toegegepast op het bootstrappen van een corpus uit verschillende tagsets. We mogen hopen dat dit betere resultaten zal opleveren dan het trainen van de losse taggers op de minimale hoeveelheid beschikbare data, aangezien ieder van de taggers een (verschillend?) deel van het probleem gedeeltelijk oplost.

Als machine learning methode hebben we Memory-Based Learning gebruikt, geïmplemeteerd in het TiMBL systeem [Daelemans *et al.* 1999]. Dit is hetzelfde leeralgoritme dat in MBT gebruikt wordt. De instanties van de taak worden in het geheugen opgeslagen als feature-value vectoren. De uitvoer van iedere tagger is een feature. Verder nemen we het te taggen woord zelf (focuswoord) mee als feature. Wanneer we nu een test-instantie aan het systeem

voorleggen dan wordt de classificatie hiervoor geëxtrapoleerd uit de dichtsbijzijnde trainingsinstanties. De gelijkenis tussen twee instanties X en Y, met feature-waarden x_i en y_i, wordt berekend op basis van de zogeheten Overlap-IG metriek (Vergelijkingen 4 en 5). De afstand tussen X en Y is een gewogen som van de afstand per feature:

$$\Delta(X,Y) = \sum_{i=1}^{n} GR(i)\delta(x_i, y_i) \tag{4}$$

Waarbij GR(i) de Gain Ratio is van feature i, zoals gegeven in formule 2, en waarin de aftand tussen twee waarden van hetzelfde feature berust op overlap:

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \tag{5}$$

Wederom doen we een 10CV voor ieder experiment, zodat ieder woord in de dataset van 2388 tokens precies één keer wordt gebruikt om te testen. In Tabel 5 staat het gemiddelde van de tien runs. Onder "alles" staat de combinatie van alle 10 de taggers: 86.3%. Daarachter staat een serie van experimenten waarbij telkens een van de taggers is weggelaten. We zien dat sommige taggers meer "nodig zijn" dan anderen. Dat sommige (D-Tale, KEPER, en de WOTAN-II taggers) individueel overbodig zijn in het geheel zien we aan het feit dat de accuraatheid licht omhoog gaat als we deze weglaten. De beste score uit dit experiment (86.6% combi–zonder D-Tale) is inderdaad aanzienlijk hoger dan de "from scratch" getrainde TnT tagger (82.7%).

					WOTAN-I			WOTAN-II		
alles	-woord	-D-Tale	-KEPER	-CORRie	-MBT	-MX	-TnT	-MBT	-MX	-TnT
86.3	85.9	86.6	86.4	86.1	86.0	85.9	86.1	86.4	86.3	86.5

Tabel 5: De accuraatheid van de combinatie van taggers met behulp van Memory-Based Learning (TiMBL). De getallen zijn een gemiddelde van een 10 voudige cross-validatie. De eerste kolom combineert alle taggers en het woord zelf; de volgende kolommen laten zien wat het effect is van het *verwijderen* van deze feature uit de combinatie: een lagere score betekent dat het meer schade aanricht om deze tagger (=feature) weg te laten.

Zulke "weglaatexperimenten" zijn ook met combinaties van features te doen, en we weten niet wat de beste features zijn voordat we alle combinaties hebben uitgeprobeerd, maar het aantal mogelijkheden is erg groot (2^{10}). Daarom hebben we een aantal voor de hand liggende combinaties uitgeprobeerd: allen de groep met minder gedetaileerde tagset (D-Tale, KEPER en CORRie), de groep met WOTAN-I, met WOTAN-II, per tagger-generator (dus alleen MBT, WOTAN-I en WOTAN-II samen, etc.), en de losse Non-WOTAN taggers[6]. Deze variaties zijn uitgevoerd met en zonder toevoeging van het focuswoord als feature (Zie Tabel 6. De combinatie van (bijna) alle taggers blijft echter superieur.

Uit Figuur 1 kunnen we een indirecte schatting maken van de afhankelijkheid van de resultaten van de hoeveelheid beschikbare data. De accuraatheid stijgt bij 100% van de data (2388 tokens) nog steeds sterk, en met een optimistische extrapolatie lijkt een score boven de 90% goed mogelijk bij een trainingset van 10000 woorden.

[6]De cijfers voor deze taggers *zonder woord* zijn uiteraard identiek aan die in Tabel 3.

	zonder woord	met woord
Non-WOTAN	62.1	80.4
D-Tale	50.6	78.9
KEPER	42.8	74.7
CORRie	42.6	77.0
WOTAN-I	76.5	83.9
WOTAN-II	79.6	81.9
MBT	–	84.7
MXPOST	–	85.1
TnT	–	85.9

Tabel 6: De accuraatheid van taggers (10CV): per tagset-blok gecombineerd, per-systeem gecombineerd, en met of zonder het te disambigueren woord zelf als een feature. Non-WOTAN = {D-Tale,KEPER,CORRie}, WOTAN = {MBT,MXPOST,TnT}. MBT = MBT-WOTAN-I,II etc.

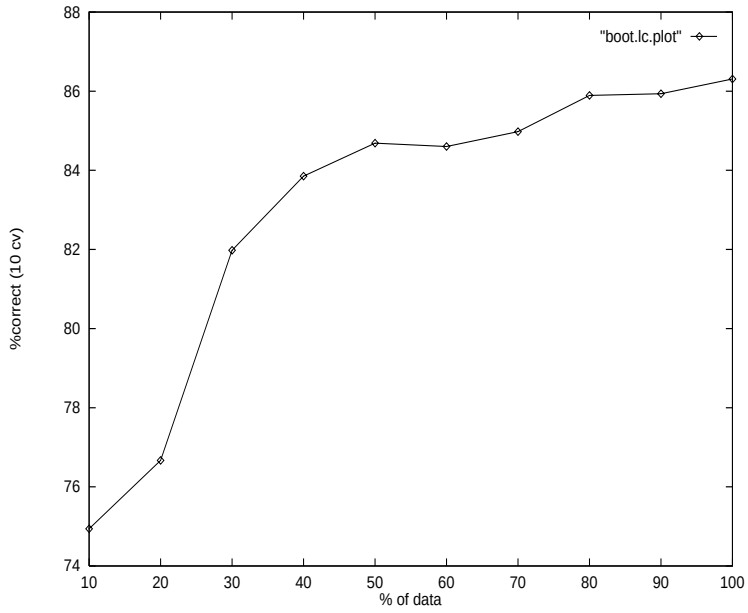
Als laatste combinatie-experiment is er nog gekeken naar de mogelijkheid om de uitvoer van het complete taggerblok (of gedeeltes ervan) op de woorden in de directe linker- en rechtercontext van het focuswoord mee te nemen als features. Dit geeft nog een lichte verbetering te zien (Tabel 7) tot 86.9%, wanneer we voor het focuswoord alle taggers en het woord zelf meenemen en voor de context alleen de WOTAN-II taggers. Uiteraard zou er langer gexperimenteerd moeten worden om een echt optimale combinatie van features te vinden. Een interessante variant is ook nog om de losse op dit corpus getrainde taggers uit Sectie 5.5 mee te nemen, en zo de trainingdata als het ware dubbel te gebruiken.

context	focus	score
alles	alles	83.0
woord+Non-WOTAN	woord+WOTAN	85.4
Non-WOTAN	woord+WOTAN	85.9
WOTAN-II	alles	86.9
WOTAN-I	alles	86.1
Non-WOTAN	alles	86.6

Tabel 7: De accuraatheid van de combinatie van taggers met de uitvoer van de taggers (in verschillende varianten) op het volgende en vorige woord als context.

6 Consequenties voor de CGN tagset

Het valt buiten de beschikbare ruimte in dit paper om een volledige foutenanalyse te doen voor de gepresenteerde systemen. Om toch een indruk te krijgen van de punten in de CGN tagset die mogelijk problemen opleveren voor de automatische taggers, geven we hier een samenvatting van de belangrijkste typen fouten van het meest accurate systeem tot nu toe, de "focus=alle-taggers,context=WOTAN-II bootstrap-tagger". Deze maakte over de gehele cross-validatie 318 fouten. Hierin zijn een aantal grotere clusters van fouten te onderscheiden:



Figuur 1: Leercurve: De accuraatheid (10CV) van de Memory-Based combinatie van taggers als een functie van het percentage van de trainingdata dat gebruikt is (100% = 2388 tokens).

- 27.% zijn fouten in de hoofdcategorie (bv. 4.4% van de fouten is verwarring tussen "ADJ(adv...)" en "BW()". Een deel hiervan betreft verkeerd opgeloste frequente ambiguïteiten. Een ander deel verkeerd gokken voor onbekende woorden. De combiner is getraind op een zeer kleine hoeveelheid data en de testset bevat dus relatief veel onbekende woorden, waaraan vaak een van de veelvoorkomende hoofdcategorieën wordt toegekend.
- Het grootste compacte cluster van fouten is de verwarring tussen hulp-, koppel-, en hoofdwerkwoorden, dat 18.8% van de totale fouten uitmaakt.
- 13.8% van de fouten zijn fouten op het feature POSITIE. De toekenning van de waarde voor dit feature hangt vaak af van de gehele structuur van de zin, bijvoorbeeld voor het onderscheid tussen predicatief en adverbiaal gebruik, terwijl de taggers slechts naar de zeer locale context kijken.
- 6.2% van de fouten bestaat uit de verwarring tussen infinitief en meervoudsvorm bij werkwoorden.
- 6% van de fouten betreft de verwarring van "fin","#neven", en "onder" voegwoordtypes.
- In 4.4% van de fouten is verkeerd gedisambigueerd tussen "VZ(init)" en "VZ(fin)".
- In 3.7% van de fouten is het feature GENUS van naamwoorden verkeerd ingevuld.
- De overige 21.1% van de fouten is verspreid over vele minder frequente types van feature mismatch.

Verder kunnen we een aantal algemene punten identificeren die samenhangen met de interactie van de tagger met de tagset:

- **De grote granulariteit van de tagset.** Statistische en andere lerende taggers hebben data nodig om hun parameters te schatten. Wanneer de tagset zeer uitgebreid is, zoals de CGN tagset tot nu toe is, dan explodeert het aantal parameters, en wordt de behoefte aan meer trainingdata exponentieel groter (teneinde vergelijkbare accuraatheden te halen). Het moet duidelijker worden in hoeverre het vanuit dit perspectief haalbaar is om op een aanvaardbaar accuraatheidsniveau te taggen met de voorgestelde tagset.

- **Niet-locale disambigueringscontext.** Aangezien taggers disambigueren op basis van een beperkte locale context, zijn er, in het algemeen, accuraatheidsproblemen te verwachten bij onderscheidingen die alleen op basis van semantiek, pragmatiek of lange-afstands afhankelijkheden op te lossen zijn. De fouten hierboven op POSITIE en WTYPE zijn hier voorbeelden van.

- **Systematische ambiguïteit.** Met name de feature POSITIE en het onderscheid tussen infinitieven en meervoudsvormen, genereren ambiguïteit voor een open klasse van woorden. We kunnen nooit verwachten dat van alle deze woorden alle instantiaties van deze ambiguïteit in de trainingsdata gezien zijn. Wanneer we deze echter systematisch aan het tagging-lexicon toevoegen met gelijke frequenties, introduceren we meer onzekerheid dan nodig is. Er moet dus gekeken worden naar de juiste kansdistributies voor deze systematische gevallen.

Verder zou er nog gekeken kunnen worden naar tagger-technieken die de features van de tagset "begrijpen" en dus niet als atomaire symbolen zien. Het onderzoek hiernaar is echter nog in een vrij experimenteel stadium [Kempe 1994, Hajic & Hladká 1998].

7 Conclusies, aanbevelingen en caveats

Naar aanleiding van een vooronderzoek zijn er een aantal taggers/lemmatizers geselecteerd om in een empirische test geëvalueerd te worden op data uit het CGN corpus. Wanneer we kijken naar de randvoorwaarden voor toepassing binnen een interactieve annotatieomgeving zijn de systemen die we getest hebben niet allemaal even geschikt. Puntsgewijs valt het volgende op te merken:

- **Lemmatisering:** Alleen D-Tale, KEPER, Xerox en MBT lemmatiseren. KEPER en MBT hanteren hierbij (vooral voor werkwoordsvormen) een andere "stijl" waardoor ze een slechte overeenstemming hebben met de handmatig aangebrachte proeflemmatisering.

- **Lexicon koppeling:** Alle geteste taggers gebruiken een eigen "tagging-lexicon" en geen van de systemen biedt de mogelijkheid om direct aan een externe lexicale database gekoppeld te worden. Maar alleen op die manier kan de "lexicale koppeling" van het corpus vergemakkelijkt worden. Voor MBT, D-Tale, KEPER, CORRie en TnT lijkt het in principe wel mogelijk om zo een verbinding tot stand te brengen. MXPOST gebruikt intern helemaal geen lexicon. Bij de Xerox-tagger lijkt het probleem vooral te liggen in de beschikbaarheid van de ontwikkelomgeving. Dit brengt ons bij het volgende punt.

- **Adaptabiliteit:** Het lijkt duidelijk wenselijk om de taggers incrementeel bij te trainen en het lexicon uit te breiden. Voor D-Tale is dit niet mogelijk omdat de taggerontwikkeling een taalkundig gemotiveerd handmatig proces is. Voor KEPER en Xerox zijn mogelijk complicaties te verwachten vanwege de beperkte openheid van de (commerciële) software.

- **Meerdere keuzes en zekerheidsmaten:** Aangezien iedere automatische tagger fouten maakt en zal blijven maken, is het belangrijk voor de correctie dat er een indicatie is op welke plaatsen de tagger onzeker is, en in dat geval (gerankte) alternatieven aan te bieden. Xerox, KEPER en D-Tale geven slechts een keuze terug, zonder zekerheidsmaat. CORRie en MXPOST zijn (eenvoudig) aan te passen om het gewenste gedrag te vertonen. MBT kan een distributie van tags binnen de "nearest neighbors" en hun mate van gelijkenis teruggeven. TnT geeft kansen van tags, gegeven het trigram Hidden Markov Model.

- **Software eisen:** Alle taggers lijken te werken met gangbare platforms (UNIX/Windows) en gangbare input/output voorwaarden (ASCII invoer via stdin of van file). Geen van de geteste taggers houdt uitgebreid rekening met SGML of andere markup coderingen.

- **Hardware eisen:** Het is niet geheel duidelijk of sommige oplossingen (zoals het bootstrappen uit 10 verschillende taggers) of het gebruik van combi-taggers geen te hoge eisen stelt aan de hoeveelheid CPU en RAM.

Wanneer we puur naar de tagset en de taggingacuraatheid van de geteste systemen en naar het gewenste niveau van prestatie kijken, worden er een aantal onzekere factoren in de overweging geïntroduceerd. Wat is een haalbare accuraatheid met deze taggers en tagset? En wat is eigenlijk de gewenste accuraatheid? We moeten immers uit de redelijk kleinschalige experimenten in dit paper gaan extrapoleren naar de productie van het complete 10M woord CGN corpus. In het begin van dit document is aangegeven dat er in feite een tweetal fases/doelen te onderscheiden zijn in de annotatie, en dat er dus ook twee soorten evaluatiecriteria te onderscheiden zijn.

Voor de eerste fase moet met de bestaande taggers snel een initieel corpus gebootstrapt worden. Gegeven de ervaring met andere corpora en tagsets moet dit initiële deel $\pm$ 50-200 duizend woorden zijn. Het lijkt in ieder geval raadzaam om met de nu opgedane ervaring verder te gaan in het handmatig annoteren van meer data op basis van slechts het lexicon (zonder automatische tagger). Op de dataset van 2388 woorden hebben we gezien dat de beste mapping vanuit een bestaande individuele tagger een accuraatheid van slechts 77.5% oplevert (TnT-WOTAN-II). Daarentegen levert het trainen van een tagger op deze data al 82.7% (wederom TnT) op. De combinatie-tagger op basis van Memory-Based Learning laat zien dat een accuraatheid van 86.3% haalbaar is. Wanneer de featureset uitbreiden naar de uitvoer van de bestaande taggers op de omliggende woorden halen we zelfs 86.9%. Het zou echter bezwaarlijk kunnen zijn (qua software en hardware) om zo veel verschillende taggers tegelijk te gebruiken. We hebben echter gezien dat de combinatie TnT-WOTAN-I, TnT-WOTAN-II, met focuswoord een acuraatheid van 85.9% haalde. Dit geeft duidelijk aan dat het incorporeren van het CGN-lexicon in het taggingproces cruciaal is. Om de verschillende delen van de combinatie met een machine-learning methode in het taggingproces te integreren zal er echter nog wel enige software-engineering inspanning geleverd moeten worden (geschatte duur: 1 à 2 weken).

Op basis van een optimistische extrapolatie lijkt het goed mogelijk om met de beste methodes (combinatie-bootstrappen) en een trainingcorpus van rond de tienduizend woorden een accuraatheid van boven de 90% te halen. Gegeven dat in [Zavrel 1999] taggingsnelheden tot duizend woorden/uur gerapporteerd zijn, is dit haalbaar met een paar weken werk[7]. Een factor waarover helaas weinig bekend is, is de mate waarin de snelheid van de menselijke taggers toeneemt met een toenemende accuraatheid van de automatische taggers. Het is dus onduidelijk waar het breekpunt ligt van investeren in meer manuele training-data voor de bootstrap tagger, zeker gegeven het feit dat we na annotatie van een paar honderduizend woorden overgaan naar de tweede fase.

In de tweede fase, wanneer er al genoeg CGN materiaal is om een eigen CGN tagger te trainen is een bootstrap methode niet meer relevant. De tagger die vanaf dat punt gebruikt wordt moet alleen de productie van de rest van het corpus efficient maken. De tagger(s) in deze tweede fase hoeven niet dezelfde te zijn als die in de eerste fase. Het ligt voor de hand om ook in deze fase TnT te gebruiken, die in de experimenten in deze paper en bij verdere in Tilburg uitgevoerde experimenten op verschillende grote (tekst)corpora (LOB, Penn Treebank, Eindhoven WOTAN-I/II) consistent de beste (of nagenoeg gelijk aan de beste) performance te zien gaf. Bovendien is TnT al geïntegreerd in de Annotate-toolkit, zodat de opbouw van het corpus en het periodiek bijtrainen van de tagger eenvoudig binnen de zelfde omgeving kan plaats vinden. Wanneer het duidelijk wordt dat met deze tagger geen bevredigende accuratesse bereikt wordt, dan kan alsnog de toevlucht worden genomen tot combinatie met andere taggers à la [Van Halteren *et al.* 1998]. In dat geval moet er nog een extra inspanning gedaan worden om de combi-tagger in de annotatie-tool te integreren.

Referenties

[Brill & Wu 1998] Eric Brill and Jun Wu 1998. Classifier Combination for Improved Lexical Disambiguation. Proc. of COLING-ACL'98, Montreal.

[Chanod & Tapanainen 1995] Jean-Pierre Chanod and Pasi Tapanainen Creating a tagset, lexicon and guesser for a French tagger. Proc. of the ACL SIGDAT workshop on "From Texts To Tags: Issues In Multilingual Language Analysis". pp. 58-64. University College Dublin, Ireland, 1995.

[Daelemans *et al.* 1999] Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch 1999. TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide ILK Technical Report 99-01, 1999. Verkrijgbaar van URL `http://ilk.kub.nl/`

[Van Eynde 1999] Frank van Eynde 1999. Part of Speech Tagging. *werkdocument Werkgroep Corpusannotatie CGN*, Versie van 10 mei 1999. Verkrijgbaar van URL `http://www.ccl.kuleuven.ac.be/CGN/tagset-mei.html`.

[GRACE] GRACE : Grammaires et Ressources pour les Analyseurs de Corpus et leur Evaluation 1998. Website op URL `http://m17.limsi.fr/TLP/grace/`

[7] In die experimenten liet echter de accuratesse te wensen over. Het is dus nog maar de vraag welke snelheid bereikbaar met behoud van zeer hoge accuraatheid.

[Hajic & Hladká 1998] Jan Hajic and Barbora Hladká 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich Structured Tagset Proc. of COLING-ACL'98, Montreal.

[Van Halteren 1999] Hans van Halteren 1999. The WOTAN2 Tagset Manual (under construction). versie van april 1999.

[Van Halteren *et al.* 1998] Hans van Halteren, Jakub Zavrel, and Walter Daelemans 1998. Improving data driven wordclass tagging by system combination . Proc. of COLING-ACL'98, Montreal.

[Karlsson *et al.* 1995] Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila (eds.) 1995. Constraint Grammar: a language-independent system for parsing unrestricted text. Volume 4 of Natural Language Processing. Mouton de Gruyter, Berlin and New York.

[Kempe 1994] Andre Kempe 1994. Probabilistic Tagging with Feature Structures Proc. of COLING-94, Kyoto. Verkrijgbaar van URL `http://xxx.lanl.gov/cmp-lg/9410027`

[Màrquez *et al.* 1998] Lluis Marquez, Lluis Padro, and Horacio Rodriguez 1998. Improving Tagging Performance by Using Voting Taggers. Proc. of Natural Language Processing & Industrial Applications (NLP+IA/TAL+AI'98), Moncton, NB, Canada, August 1998.

[Nivre *et al.* 1996] Joakim Nivre, Leif Grönqvist, Malin Gustafsson, Torbjörn Lager, and Sylvana Sofkova 1996. Tagging Spoken Language Using Written Language Statistics. Proc. of COLING-96. Verkrijgbaar van URL `http://www.ling.gu.se/~joakim/`.

[Plähn 1998] Oliver Plaehn 1998. Annotate. Bedienungsanleitung. *Technical Report, Dept. Computerlinguistik, Universität de Saarlandes*, 3 april 1998. Verkrijgbaar van URL `http://www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html`.

[Quinlan 1993] J.R. Quinlan. C4.5*: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[Samuelsson 1996] Christer Samuelsson 1996. Handling Sparse Data by Successive Abstraction. CLAUS Technical Report No. 69, Universität de Saarlandes.the Saarland, Germany.

[Zavrel 1999] Jakub Zavrel 1999. Annotator-overeenstemming bij het manuele tagging-experiment. *werkdocument Werkgroep Corpusannotatie CGN.* Verkrijgbaar van URL `http://ilk.kub.nl/~zavrel/agreement-verslag.ps`.

A Tagsets

A.1 INL tagset (basiscategorieën)

De INL tagset, die gebruikt wordt door de CORRie tagger van Theo Vosse, is de meest eenvoudige tagset (11 tags) van de hier beschouwde. Hij maakt alleen onderscheid tussen de hoofdcategorieën (het feature POS in de CGN tagset). Van de overige features van de CGN tagset worden slechts de feature NTYPE (eigen, soort) onderscheiden en er is een grove onderverdeling van de werkwoorden.

```
p = persoonlijk voornaamwoord
w of een cijfer = werkwoord
v = voorzetsel
e = eigennaam
l = lidwoord
b = bijwoord
a = bijvoegelijk naamwoord
z = zelfstandig naamwoord
t = telwoord
c = voegwoord
o = overig
```

A.2 KEPER

De KEPER tagset bestaat uit 24 tags, en is volgens Polderland BV. ontworpen met het oog op Information Retrieval toepassingen. Een van de meest opvallende eigenschappen van deze tagset is dat er geen aparte categorie is voor telwoorden. Uitgechreven telwoorden vallen onder de naamwoorden, hun numerieke equivalenten onder NUM. Verder worden alle hoofklasses (POS) onderscheiden en zijn er acht vertakkingen van de werkwoordsklasse.

```
*               idiosyncratic [dwz. het woord is zijn eigen tag,
       zoals het to-infinitive]
ADJ             adjective
ADV             adverb
AUX INF         auxiliary (infinitive)
AUX PCP1        auxiliary (present participle)
AUX PCP2        auxiliary (past participle)
AUX SG VFIN     auxiliary (past tense)
CC              coordinator
CS              subordinator
DET             determiner
INFMARK>        infinitive marker (idiosyncratic)
INTERJ          interjection
N ABBR          abbreviation
N NOM PL        noun (plural)
N NOM SG        noun (singular)
NUM             number
PCP1            verb (present participle)
PCP2            verb (past participle)
PMK             punctuation mark
PREP            preposition
PRON            pronoun
Q               quantifier
V INF           verb (infinitive)
V SG VFIN       verb (past tense)
```

A.3 D-Tale

De tagset van D-Tale, ontwikkeld door de afdeling lexicografie van de VU, bestaat uit 45 tags. Er worden distincties gemaakt op ruwweg het niveau van de POS en subtype.

Tagset:

categorie	specificatie	uitleg
verb	pres sg	present singular
	pres pl	present plural
	past sg	past singular
	past pl	past plural
	inf	infinitive
	papa	past participle
	prespa	present participle
	inf/pl	infinitive/ present plural
	*kop	verb met koppelteken
noun	sg	singular
	pl	plural
	prop	proper
	abbr	abbreviation
	*kop	noun met koppelteken
pron	pers	personal
	poss	possessive
	demo	demonstative
	rel	relative
	refl	reflexive
	quest	interrogative
num	ord	ordinal
	card	cardinal
det	art	article
	poss	possessive
	demo	demonstative
	indef	indefinite
	refl	reflexive
	quest	interrogative
adj	--	
	comp	comparative
	super	superlative
	*kop	adj met koppelteken
	abbr	abbreviation

adv	--	
	abbr	abbreviation
	*kop	adverb met koppelteken
prep	--	preposition
part	--	particle
int	--	interjection
conj	--	conjunction
	subo	subordinate
	coor	coordinate
punct	--	interpunction, end-of-sentence
symb	--	symbol
	--	multi word unit

A.4 XEROX

De tagset van de Nederlandse XEROX tagger bestaat uit 49 tags en maakt, naast onderscheiden in hoofdklasse en subtype, bij sommige POS ook de voor de CGN tagset relevante functionele onderscheiden (bv.: attributief vs. predicatief of adverbiaal gebruik van adjectieven, preposities en postposities). Er wordt niet of nauwelijks geclassificeerd op basis van morfologische (persoon, getal) of lexicale features (eigen- vs. soortnamen, etc.).

tag	description	example
NOUN	Common Noun or	[de] hoed; [het] goede
	Proper Name	Peter; [de] Betuwelijn
ADJA	Attributive Adjective	[een] snelle auto
ADJD	Adverbial or Predicative Adjective	[hij rijdt] snel
PADJ	Postmodifying Adjective	[wat] aardigs
VVFIN	Finite Substantive Verb	[hij] zegt
VVINF	Infinitive Substantive Verb	[hij zal] zeggen
VVPP	Past Participle Substantive Verb	[hij heeft] gezegd
VAFIN	Finite Auxiliary Verb	[hij] is [geweest]
VAINF	Infinitive Auxiliary Verb	[hij zal] zijn
VAPP	Past Participle Auxiliary Verb	[hij is] geweest
VPRES	Present Participle Verb	[dit] zeggend
ART	Article	een [bus], het [busje]
ADV	Non-Ajectival Adverb	[hij rijdt] vaak
PROADV	Pronominal Adverb	[hij praat] hierover
WADV	Interrogative Adverb	waarom [gaat hij]

CWADV	Interrogative Adverb or	wanneer [gaat hij weg]
	Subordinate Conjunction	wanneer [hij nu weggaat]
CARD	Cardinals	125, vijf, 12/2
ORD	Ordinals	vijfde, 125ste, 12de
PREP	Preposition	[hij is] in [het huis]
POSTP	Postposition	[hij liep zijn huis] in
CIRCP	Right Part of Circumposition	[hij viel van het dak] af
CMPDPART	Right Truncated Part of Compound	honde- [en kattevoer]
PERS	Personal Pronoun	hij [sloeg] hem
POSDET	Possessive Pronoun	mijn [boek]
DEMPRO	Demonstrative Pronoun	deze [gaat goed]
DEMDET	Demonstrative Determiner	deze [machine gaat goed]
INDPRO	Indefinite Pronoun	beide [gingen weg]
INDPRE	Indefinite Predeterminer	al [de broers]
INDDET	Indefinite Determiner	geen [broer]
INDPOST	Indefinite Postdeterminer	[de] beide [broers]
RELPRO	Relative Pronoun	[de man] die [lachte]
RELSUB	Relative Conjunction	[Het kind] dat ... [Het feit] dat ...
WPRO	Interrogative or	[de vraag] wie ...
	Relative Pronoun	[de man] wie ...
WDET	Interrogative or	[de vraag] wier [man] ...
	Relative Determiner	[de vrouw] wier [man] ...
CON	Co-ordinating Conjunction	[Jan] en [Marie]
SUBCON	Subordinating Conjunction	Hoewel [hij er was]
INFCON	Infinitive Conjunction	om [te vragen]
COMCON	Comparative Conjunction	[zo groot] als
		[groter] dan
PTKTE	Infinitive Particle	[hij hoopt] te [gaan]
PTKA	Adverb Modification	[hij wil] te [snel]
PTKNEG	Negation	[hij gaat] niet [snel]
PTKVA	Separated Prefix, Pronominal Adverb	[daar niet] mee
	or Verb	[hij loopt] mee
FM	Foreign Material	[wat een] crime [!]
ITJ	Interjections	jawel, och, ach
XY	Residual	\^C5, wer34a
CM	Comma	,
SENT	Sentence Final Punctuation	; . ?
PUNCT	Other Punctuation	‘‘~’’ ‘~ ’ [~] <~ > - --
SGML	SGML-tag	<article>

A.5 WOTAN-I

De WOTAN tagset en zijn opvolger, de WOTAN-II tagset, ontwikkeld aan de Katholieke Universiteit Nijmegen lijken van de beschouwde tagsets het meest op de CGN tagset. Zowel qua granulariteit (enkele honderden tags), als qua formaat (feature-value structuren), als qua morfologische en lexicale verfijdheid. Hieronder geven we de opsomming van de WOTAN-I tagset. De WOTAN-II tagset is gedocumenteerd in [Van Halteren 1999].

In de onderstaande opsomming zijn telkens per hoofdcategorie de features (posities) en hun mogelijke waarden gegeven. (Optionele posities zijn omgeven door vierkante haken):

```
N                       Zelfstandig naamwoorden
    pos. 1
        soort           soortnamen
        eigen           eigennamen
    pos. 2
        ev              enkelvoud
        mv              meervoud
    pos. 3
        neut            naamval
        gen             genitief
        dat             datief

Pron                    Voornaamwoorden
    pos. 1
        per             persoonlijk
        bez             bezittelijk
        ref             reflexief
        rec             reciprook
        aanw            aanwijzend
        betr            betrekkelijk
        vrag            vragend
        onbep           onbepaald
    pos. 2 (per;bez;ref)
        1               1e persoon
        2               2e persoon
        3               3e persoon
    pos. 3 (per;bez;ref)
        ev              enkelvoud
        mv              meervoud
        ev_of_mv        geen onderscheid tussen enkel- of meervoud
    pos. 4 (per;bez)
    pos. 2 (vrag;aanw;onbep;rec;betr)
        neut            geen naamval
        nom             nominatief
        gen             genitief
        acc             accusatief
        dat_of_acc      geen onderscheid tussen datief
                        of accusatief
    pos. 5 (bez)
    pos. 3 ((vrag;aanw;onbep;betr)
        attr            attributief gebruikt
        zelfst          zelfstandig gebruikt
```

```
   [pos. 4 (aanw)]
       [w_eigen          ‘‘eigen’’]
       [w_zelf           ‘‘zelf’’]

Art                      Lidwoorden
    pos.1
        bep              bepaalde lidwoorden
        onbep            onbepaalde lidwoorden
    pos. 2
        zijd             zijdig
        zijd_of_mv       zijdig of meervoud
        onzijd           onzijdig
        onzijd_of_mv     onzijdig of meervoud
    pos. 3
        neut             geen naamval
        gen              genitief
        dat              datief

Adj                      Bijvoeglijke Naamwoorden
    pos.1
        adv              adverbiaal gebruikt
        attr             attributief gebruikt
        zelfst           zelfstandig gebruikt
    pos. 2
        stell            stellende trap
        vergr            vergrotende trap
        overtr           overtreffende trap
    pos. 3
        onverv           onvervoegde vorm
        verv_neut        gewone vervoegde vorm
        verv_gen         genitief vorm
        verv_mv          meervoudsvorm

Adv                      Bijwoorden
    pos.1
        gew              gewone bijwoorden
        pron             voornaamwoordelijke bijwoorden
        deel_v           bijwoordelijk of prepositioneel deel
                         van gescheiden samengesteld werkwoord
        deel_adv         prepositioneel deel van gescheiden
                         voornaamwoordelijk bijwoord
    pos. 2 (gew;pron)
        geen_func        niet voorzien van functie-informatie
        betr             functie betrekkelijk
        vrag             functie vragend
        aanw             functie aanwijzend
        onbep            functie onbepaald
```

```
        er                  ‘‘er’’
    pos. 3 (gew,geen_func,...)
        stell               stellende trap
        vergr               vergrotende trap
        overtr              overtreffende trap
    pos. 4 (gew,geen_func,...)
        onverv              onvervoegde vorm
        verv_neut           gewone vervoegde vorm

Num                         Telwoorden
    pos.1
        hoofd               hoofdtelwoorden
        rang                rangtelwoorden
    pos. 2
        bep                 bepaalde telwoorden
        onbep               onbepaalde telwoorden
    pos. 3
        zelfst              zelfstandig gebruikt
        attr                attributief gebruikt
    pos. 4 (hoofd,onbep,...)
        stell               stellende trap
        vergr               vergrotende trap
        overtr              overtreffende trap
    pos. 4,5 (rang,onbep,...)
        onverv              onvervoegde vorm
        verv_neut           gewone vervoegde vorm
        verv_gen            genitief vorm
        verv_mv             meervoudsvorm

V                           Werkwoorden
    pos. 1
        trans               transitieve werkwoorden
        refl                reflexieve werkwoorden
        intrans             intransitieve werkwoorden
        hulp                hulpwerkwoorden
        hulp_of_kopp        hulp- of koppelwerkwoorden
    pos. 2
        ott                 onvoltooid tegenwoordige tijd
        ovt                 onvoltooid verleden tijd
        teg_dw              tegenwoordig deelwoord
        verl_dw             verleden deelwoord
        inf                 infinitief
        conj                conjunctief
        imp                 imperatief
    pos. 3 (ott;ovt)
        1                   1e persoon
        2                   2e persoon
```

```
        3                   3e persoon
        1_of_2_of_3         mv-vorm ott, ev- en mv-vorm ovt
    pos. 4 (ott;ovt)
        ev                  enkelvoud
        mv                  meervoud
   [pos. 3 (inf)]
       [subst               substantivaal gebruikt]
    pos. 3 (teg_dw;verl_dw)
        onverv              onvervoegde vorm
        verv_neut           gewone vervoegde vorm
        verv_gen            genitief vorm
        verv_mv             meervoudsvorm

Prep                        Voorzetsels
    pos. 1
        voor                echte voorzetsels
        achter              achterzetsels
        comb                gecombineerd voorzetsel
        voor_inf            ‘‘te’’ (voor infinitief)

Conj                        Voegwoord
    pos. 1
        neven               nevenschikkende voegwoorden
        onder               onderschikkende voegwoorden
    pos. 2 (onder)
        met_fin             voegwoord gevolgd door finiete bijzin
        met_inf             voegwoord gevolgd door infiniete bijzin

Int                         Tussenwerpsels
    geen subposities

Punc                        Leestekens
    pos. 1
        aanhaal_dubb        dubbele aanhalingstekens
        aanhaal_enk         enkele aanhalingstekens
        dubb_punt           dubbele punt
        en                  ampersand (‘‘&’’)
        ged_streepje        gedachten streepje (‘‘-’’
        haak_open           haakje openen
        haak_sluit          haakje sluiten
        hellip              horizontale ellipsis (‘‘...’’)
        is_gelijk           is-gelijk-teken
        komma               komma
        ligg_streep         underscore (‘‘_’’)
        maal                maal-teken (‘‘x’’)
        plus                plus-teken
        punt                punt
```

```
        punt_komma          puntkomma
        schuin_streep       slash (``/'')
        uitroep             uitroepteken
        vraag               vraagteken

Misc                        Diversen
    pos. 1
        afkort              niet benoemde afkortingen
        vreemd              vreemde expressie (buitenlands e.d.)
        symbool             niet bij Punc inbegrepen symbolen
```