Die ontwikkeling van 'n woordafbreker en kompositumanaliseerder vir Afrikaans

S. Pilon, M.J. Puttkammer & G.B. van Huyssteen
Sentrum vir Tekstegnologie (CTexT)
Potchefstroomkampus
Noordwes-Universiteit
POTCHEFSTROOM
E-pos: Sulene.Pilon@nwu.ac.za
Martin.Puttkammer@nwu.ac.za
Gerhard.VanHuyssteen@nwu.ac.za

Abstract

The development of a hyphenator and compound analyser for Afrikaans

The development of two core-technologies for Afrikaans, viz. a hyphenator and a compound analyser is described in this article. As no annotated Afrikaans data existed prior to this project to serve as training data for a machine learning classifier, the core-technologies in question are first developed using a rule-based approach. The rule-based hyphenator and compound analyser are evaluated and the hyphenator obtains an f-score of 90,84%, while the compound analyser only reaches an f-score of 78,20%. Since these results are somewhat disappointing and/or insufficient for practical implementation, it was decided that a machine learning technique (memory-based learning) will be used instead. Training data for each of the two core-technologies is then developed using "TurboAnnotate", an interface designed to improve the accuracy and speed of manual annotation. The hyphenator developed using machine learning has been trained with 39 943 words and reaches an f-score of 98,11% while the f-score of the compound analyser is 90,57% after being trained with 77 589 annotated words. It is concluded that machine learning (specifically memory-based learning) seems an appropriate approach for developing core-technologies for Afrikaans.

Opsomming

Die ontwikkeling van 'n woordafbreker en kompositum-analiseerder vir Afrikaans

In hierdie artikel word die ontwikkeling van twee kerntegnologieë vir Afrikaans, ’n woordafbreker en ’n kompositumanaliseerder, beskryf. Aangesien geen geannoteerde data waarmee masjienleermodules afgerig kan word voor hierdie projek beskikbaar was nie, word eers van ’n reëlgebaseerde benadering gebruik gemaak om hierdie kerntegnologieë te ontwikkel. Die reëlgebaseerde modules word geëvalueer en die woordafbreker behaal ’n f-telling van 90,84% en die kompositumanaliseerder ’n f-telling van 78,20%. Aangesien hierdie resultate nie heeltemal bevredigend vir praktiese implementering is nie, word ’n masjienleertegniek (geheuegebaseerde leer) vervolgens gebruik om hierdie modules te ontwikkel. Afrigtingsdata vir albei die kerntegnologieë word ontwikkel met behulp van “Turbo-Annotate”, ’n koppelvlak wat ontwikkel is om die akkuraatheid en spoed van handmatige annotasie te verhoog. Die masjienleerwoordafbreker word afgerig met 39 943 geannoteerde woorde en behaal ’n f-telling van 98,11%, terwyl die kompositumanaliseerder’n f-telling van 90,57% behaal nadat dit met 77 589 geannoteerde woorde afgerig is. Dit word ten slotte gestel dat masjienleer (spesifiek geheuegebaseerde leer) suksesvol blyk te wees in die ontwikkeling van kerntegnologieë vir Afrikaans.

1. Inleiding

Die groei en ontwikkeling van ’n mensetaaltegnologie-industrie van ’n taal is afhanklik van die ontwikkeling van kerntegnologieë (d.i. modules wat vir spesifieke take ontwikkel word en dan in toepassings geïmplementeer kan word) vir dié betrokke taal. Dit is daarom van kardinale belang om effektiewe en herbruikbare kerntegnologieë vir tale met ontluikende mensetaaltegnologie-industrieë te ontwikkel.

Een van die belangrikste kerntegnologieë in die ontwikkeling van taaltegnologietoepassings is ’n outomatiese morfologiese analiseerder (d.i. ’n module wat gebruik word om woorde en hulle konstituente outomaties te analiseer; Lezius *et al.*, 1996; Minnen *et al.*, 2001; Van den Bosch & Daelemans, 1999). Morfologiese analiseerders word nie alleen in teksgebaseerde toepassings (soos spel- en grammatikatoetsers, masjienvertaal-, tekskategorisering- en inligtingonttrekkingsisteme) gebruik nie, maar ook in spraakgebaseerde toepassings (byvoorbeeld spraakherkenningsisteme en dialoogsisteme; vgl. Sproat, 1992:2-14). Aangesien morfologiese analise es-

sensieel is in die meeste taaltegnologietoepassings (Daelemans *et al.*, 2005), is dit daarom van kernbelang om ’n gesofistikeerde, herbruikbare morfologiese analiseerder vir ’n taal soos Afrikaans te ontwikkel.

Alvorens sodanige morfologiese analiseerder ontwikkel word, moet deeglik besin word oor die soort analises wat die analiseerder moet kan doen. Dit word gewoonlik deur die aard van die toepassing waarin die analiseerder gebruik gaan word, bepaal (Sproat, 1992:2). Indien die analiseerder byvoorbeeld vir outomatiese opsomming (*summarisation*) gebruik gaan word, is stamidentifisering (*stemming*) van minder belang, terwyl dit juis weer essensieel is vir dokumentonttrekking (*document retrieval*). In teks-na-spraaksisteme is lettergreepverdeling en woordsegmentering belangriker as ’n volledige morfologiese analise, wat weer juis noodsaaklik sou kon wees in ’n grammatikatoetser. Om te verseker dat die morfologiese analiseerder wat ontwikkel word, herbruikbaar is (d.i. geskik vir gebruik in ’n verskeidenheid toepassings), moet gepoog word om funksionaliteite in sodanige analiseerder in te bou wat dit in soveel moontlik toepassings bruikbaar sou kon maak.

In hierdie artikel word die ontwikkeling van ’n woordafbreker en ’n kompositumanaliseerder[1] vir Afrikaans beskryf; albei kan beskou word as kerntegnologieë wat in ’n outomatiese morfologiese analiseerder geïmplementeer kan word. Van ’n woordafbreker word verwag om meerlettergrepige woorde in lettergrepe te verdeel (bv. om *fakulteitsraad* te analiseer as *fa-kul-teits-raad*), terwyl ’n kompositumanaliseerder komposita in woorddele en valensiemorfeme moet verdeel (bv. om *fakulteitsraad* te analiseer as bestaande uit die twee woorde *fakulteit* en *raad*, plus die valensiemorfeem *-s-*; met ander woorde om as afvoer *fakulteit_s+raad* te lewer).

Verder moet ook besin word oor die metodes wat gebruik gaan word om die analiseerder te ontwikkel. In die literatuur word gewoonlik onderskei tussen linguistiese/reëlgebaseerde en datagedrewe/statistiese/stogastiese metodes en benaderings (vgl. Jurafsky & Martin, 2000; Voutilainen, 1999). In linguistiese benaderings word meestal gebruik gemaak van handgemaakte reëls wat gebaseer is op die taalkundige kennis van die ontwikkelaar, grammatikas van die taal, voorbeelde uit korpora, woordeboeke, ensovoorts. Sodanige reëls

1 Die outeurs verstaan onder *analiseer* (*analyse*) die verdeling van ’n kompositum in konstituente en stel dit teenoor *ontleed* (*parse*) wat verwys na ’n geannoteerde en gedetailleerde analise.

kan op 'n verskeidenheid maniere geïmplementeer word, waarvan die bekendste die gebruik van eindigetoestandmodelle (*finite-state models*; Koskenniemi, 1983) is. Linguistiese benaderings tot morfologiese analise lewer gewoonlik 'n "deep, accurate, and complete analysis and understanding of text", maar vereis ook "an inordinate amount of time, effort, and … cost" (Text Analysis International, 2001).

Datagedrewe benaderings, daarenteen, is gewoonlik aansienlik goedkoper, makliker en vinniger. Aangesien statistiese modelle geskep word deur die outomatiese analise van korpora (dikwels deur gebruik te maak van masjienleertegnieke) word groot hoeveelhede (geannoteerde) data vereis. Die ontwikkeling van sodanige data kan egter ook tydrowend en duur wees, veral wanneer van gekontroleerde leer (*supervised learning*) gebruik gemaak word. Statistiese modelle kan op 'n verskeidenheid maniere geïmplementeer word, waaronder kollokasiematryse, Markovmodelle, lokale reëls, neurale netwerke, ensovoorts (Voutilainen, 1999:9).

In die volgende afdeling word eers gefokus op die ontwikkeling en evaluasie van 'n reëlgebaseerde woordafbreker en daarna op dié van 'n reëlgebaseerde kompositumanaliseerder. In Afdeling 3 word die ontwikkeling van 'n datagedrewe woordafbreker en kompositumanaliseerder beskryf en die resultate wat dié modules in evaluasies behaal het, bespreek. Die artikel sluit af met aanbevelings ten opsigte van toekomstige werk wat kan lei tot die verbetering van die datagedrewe modules.

2. Reëlgebaseerde benadering

Aangesien daar geen geannoteerde afrigtingsdata vir Afrikaans beskikbaar was by die aanvang van dié projek nie, is besluit om 'n reëlgebaseerde benadering in die ontwikkeling van hierdie kerntegnologieë te volg. Die reëlgebaseerde benadering is al in die verlede suksesvol gebruik om verskeie kerntegnologieë vir ander tale te ontwikkel. Woordafbrekers wat met behulp van reëlgebaseerde metodes ontwikkel is, sluit in dié van Boot (1984), Daelemans (1989), Liang (1983), Tutelaers (1993) en Nunn (1999). Hierdie woordafbrekers is binne verskillende kontekste ontwikkel en daarom ook op verskillende, meestal onvergelykbare maniere, geëvalueer. Nunn (1999) ontwikkel byvoorbeeld 'n woordafbreker om deel te vorm van 'n woordeboekdatabasis, en daarom moet alle moontlike afbreekplekke binne 'n betrokke woord geïdentifiseer word. Daelemans (1989) daarenteen, ignoreer afbreekplekke wat twee letters weg van

die woordgrense (d.i. die begin en einde van die woord) is, ter wille van tipografiese redes.

Met betrekking tot kompositumanaliseerders gebruik Schiller (2005) byvoorbeeld geweegde eindigetoestandoorvormers (*weighted finite state transducers*) om ’n kompositumanaliseerder vir Duits te ontwikkel. Die sisteem neem as toevoer die afvoer van ’n proses waartydens alle moontlike analises van ’n betrokke kompositum gegenereer is. Dit is die taak van die geweegde eindigetoestandoorvormer om te bepaal watter een van die moontlike analises die korrekte een is. Die presisie van hierdie sisteem lê tussen 89% en 98% en die herroeping tussen 98% en 99%. Vandeghinste (2002) gebruik ’n vorm van langstestringpassing in kombinasie met statistiese parameters om komposita in ’n Nederlandse spraakherkenningsisteem te analiseer en die resultate van die evaluasie van hierdie sisteem toon dat dit in 94,5%-98,5% van gevalle daartoe in staat is om te bepaal of woorde uit een, twee of drie konstituente bestaan.

Uit bogenoemde blyk dit dat die reëlgebaseerde benadering in verskillende kontekste suksesvol aangewend is om woordafbrekers en kompositumanaliseerders te ontwikkel. In die volgende afdelings word die ontwikkeling en evaluasie van ’n reëlgebaseerde woordafbreker en kompositumanaliseerder vir Afrikaans beskryf. Daarna sal verduidelik word waarom daar besluit is om die reëlgebaseerde benadering te laat vaar en eerder op ’n datagedrewe benadering te fokus.

2.1 ’n Reëlgebaseerde woordafbreker

Die reëlgebaseerde woordafbreker is deeltyds oor ’n tydperk van ongeveer twee jaar ontwikkel en bestaan uit 1 010 reëlmatige uitdrukkings. Hierdie reëlmatige uitdrukkings is in ’n Perl-omgewing geïmplementeer en is gebaseer op die lettergreepverdelingsbeginsels wat in die *Afrikaanse Woordelys en Spelreëls* (Suid-Afrikaanse Akademie vir Wetenskap en Kuns, 2002) vervat is. Die *Afrikaanse Woordelys en Spelreëls* bepaal byvoorbeeld in reël 1.10 dat ’n woord tussen twee identiese medeklinkers wat tussen klinkers staan, afgebreek kan word. Die reëlmatige uitdrukking in Perl sien soos volg daar uit:[2]

2 $M en $K is onderskeidelik een element uit voorafgedefinieerde lyste van alle moontlike medeklinkers ($M) en alle moontlike klinkers ($K) wat in Afrikaans kan voorkom.

```
if ($token=~/^(.*)($K)($M)($M)($K)(.*)$/) {
        if ($3 eq $4) {
                $token = $1.$2.$3.*.$4.$5.$6;
        }
 }
```

Hierdie reël impliseer dat die woord *balle* tussen die twee l'e soos volg afgebreek word: *bal*le*, waar die asterisk die moontlike afbreekplek aandui.[3]

Die reëlgebaseerde woordafbreker wat hier ontwikkel is, is met dieselfde datastel geëvalueer wat gebruik is om die masjienleerwoordafbreker af te rig (sien 3.2.1 vir inligting oor die datastel). In die evaluasie is die woordafbreker se presisie, herroeping en *f*-telling (Van Rijsbergen, 1979) met betrekking tot afbreekplekke eerstens bereken. Met ander woorde, daar is bepaal hoeveel van die potensiële afbreekplekke reg voorspel is. Die akkuraatheid van die woordafbreker op woordvlak is ook vervolgens bepaal – met ander woorde, vir hoeveel van die woorde is alle potensiële afbreekplekke reg voorspel. Die woordafbreker behaal 'n *f*-telling van 90,84% (met presisie van 92,03% en herroeping van 89,69%) op afbreekplekke, en op woordvlak behaal die reëlgebaseerde woordafbreker 'n akkuraatheid van 73,56%.

Een van die groot probleme in die ontwikkeling van hierdie module was die ordening van die reëls. Dit maak sin om die reëls volgens die vlak van spesifiekheid te rangskik, met die mees spesifieke reëls eerste. Sodoende kan uitsonderings, wat meestal met sulke spesifieke reëls hanteer word, vroeg in die proses al hanteer word. So byvoorbeeld sal die reël wat bepaal dat daar 'n afbreekplek voorkom tussen die karakterstringe *mens* en *-lik* heelwat vroeër voorkom as die reël wat bepaal dat 'n woord afgebreek kan word tussen die konsonante *n* en *s* (soos byvoorbeeld in die woorde *men*se* en *on*sin*). Dit is egter soms moeilik om te bepaal of 'n reël meer of minder spesifiek as 'n ander een is: vergelyk die laasgenoemde reël wat bepaal dat 'n woord afgebreek kan word tussen die konsonante *n* en *s*, met een wat bepaal dat daar eers ná die karakters *ns* afgebreek kan word (soos in die woorde *dans*uit*stap*pie* en *lens*in*stel*lings*). Tydens die eksperimentele ontwikkelingsfase het dit geblyk dat die ordening van die reëls 'n groot invloed het op die

3 Daar is besluit om asteriske, en nie koppeltekens nie, te gebruik om afbreekplekke aan te dui, aangesien koppeltekens soms in Afrikaanse woorde voorkom (soos byvoorbeeld *zebra-agtig*).

akkuraatheid van die module; die skuif van ’n enkele reël kan ’n merkbare, en soms selfs drastiese invloed hê op die akkuraatheid van die woordafbreker.

’n Ander probleem was die hoë frekwensie van uitsonderings – veral waar komposita ter sprake kom. So byvoorbeeld sal die reël wat bepaal dat daar tussen *n* en *ge* afgebreek moet word die woord *sangerkenning* verkeerdelik afbreek as *san*ger*ken*ning* terwyl dit eintlik *sang*er*ken*ning* behoort te wees. Vergelyk ook die verwarring wat die valensiemorfeem veroorsaak in voorbeelde soos *seunskoen* en *seunsklere.* In die eerste voorbeeld moet die woord afgebreek word as *seun*skoen*, terwyl die tweede voorbeeld afgebreek moet word as *seuns*klere.*[4]

2.2 ’n Reëlgebaseerde kompositumanaliseerder vir Afrikaans

Aangesien samestelling ’n produktiewe woordvormingsproses in Afrikaans is, moet ’n morfologiese analiseerder vir Afrikaans ’n komponent bevat wat samestellings (komposita) kan analiseer. In ’n eerste poging om konstituentgrense in komposita te identifiseer, is ’n module wat van langstestringpassing (LSP) gebruik maak, ontwikkel.

Die module soek aan die linker- en regterkant van ’n woord die langste karakterstring wat deel van die leksikon van *Afrikaanse Speltoetser 3.0* (CTexT, 2005) is. Die soektog word éérs van die regterkant van die woord af gedoen en daarna van die linkerkant van die oorblywende karakterstring af. In ’n samestelling soos *vakansiebestemming* sal die string *bestemming* dus eerste gevind word, gevolg deur die string *vakansie*. Om valensiemorfeme en woorde wat uit meer as twee konstituente bestaan te hanteer, spesifiseer die algoritme dat die karakterstring wat oorbly nadat die langste string aan die linker- en regterkant gevind is, óf deel moet wees van ’n lys van moontlike valensiemorfeme, óf ook deel moet wees van die speltoetserleksikon (vgl. Van Huyssteen & Van Zaanen, 2003).

Die LSP-kompositumanaliseerder is geëvalueer met die datastel wat gebruik is om die masjienleerkompositumanaliseerder (vgl. 3.3.1 vir

4 Hierdie probleem sou in die toekoms opgelos kon word deur ’n woord eers deur die kompositumanaliseerder te laat analiseer om potensiële woordgrense, wat meestal op lettergreepgrense dui, te identifiseer. Aangesien die modules hier as aparte, selfstandige kerntegnologieë beskryf word, word dit nie hier gekombineer om probleme soos hierdie op te los nie.

inligting oor die data) af te rig. Presisie, herroeping en *f*-telling met betrekking tot konstituentgrense (m.a.w. of die kompositumanaliseerder kan bepaal of 'n betrokke posisie in 'n woord 'n konstituentgrens is en boonop korrek kan voorspel watter konstituentgrens in die betrokke posisie voorkom), en akkuraatheid op woordvlak (m.a.w. of alle moontlike konstituentgrense in 'n woord korrek geïdentifiseer is) word bereken. Die kompositumanaliseerder behaal 'n *f*-telling van 78,20% (met presisie van 96,84% en herroeping van 65,57%) op konstituentgrense, en op woordvlak behaal die analiseerder 'n akkuraatheid van 66,40%.

Die feit dat hierdie module nie-diskriminerend is, veroorsaak 'n wesenlike probleem binne die konteks van speltoetsers, aangesien die LSP-module soms spelfoute analiseer as gangbare komposita. Dit analiseer byvoorbeeld die verkeerd gespelde **deurgans* verkeerdelik as *deur* + *gans,* **balansseer* as *balans* + *seer* en **dieman* as *die* + *man*. In 'n toepassing soos 'n speltoetser is sodanige foute uiteraard onaanvaarbaar, aangesien bogenoemde drie foutiewe woorde, nadat dit geanaliseer is, as korrek gespelde woorde deur 'n speltoetser aanvaar sal word.

Die lae herroeping van die LSP-module kan toegeskryf word aan die feit dat dit gulsig (*greedy*) is en die aard van die leksikon wat die module gebruik. Die module identifiseer telkens die langste moontlike karakterstring wat dit in die leksikon vind, en aangesien die leksikon vir speltoetserdoeleindes ontwikkel is, bevat dit heelwat morfologies komplekse woorde wat 'n groot hoeveelheid komposita insluit. Wanneer die LSP-module byvoorbeeld met die woord *dakverfwinkel* gekonfronteer word, word dit geanaliseer as *dak* + *verfwinkel* omdat die woord *verfwinkel* by die speltoetserleksikon ingesluit is. Die module sou dus in die toekoms verbeter kon word deur 'n stamlys, wat geen komposita bevat nie, as leksikon te gebruik.

Die feit dat die LSP-module net komposita kan analiseer wat uit 'n maksimum van drie komponente bestaan, is 'n verdere tekortkoming van hierdie module. Die woord *fakulteitsraadbyeenkoms* kan byvoorbeeld nie deur hierdie module geanaliseer word nie, aangesien dit uit ses konstituente bestaan (*fakulteit _ s* + *raad* + *by* + *een* + *koms*).

2.3 Gevolgtrekking

Nadat die reëlgebaseerde modules geëvalueer is, is gevind dat dit oor die algemeen nie bevredigende resultate lewer nie (in die geval van die woordafbreker), of oorveralgemeen (in die geval van die

kompositumanaliseerder) en dus nie geskik is om in toepassings soos spel- en grammatikatoetsers of in ander kerntegnologieë soos morfologiese analiseerders geïmplementeer te word nie. Alhoewel sommige van die problematiese aspekte wat hierbo bespreek is moontlik deur ’n ander reëlgebaseerde metode (bv. eindigetoestand-outomate) opgelos sou kon word, is besluit om met ’n datagedrewe benadering te eksperimenteer, eerder as om die reëlgebaseerde benadering verder te ondersoek. Hierdie besluit is hoofsaaklik gegrond op die feit dat die ontwikkeling en verbetering van reëlgebaseerde modules besonder tyd- en werksintensief is (dit vereis die kundigheid van taaleksperts, wat dikwels duur is). Dit beteken egter nie noodwendig dat metodes binne die reëlgebaseerde benadering nie vir die ontwikkeling van kerntegnologieë vir Afrikaans geskik is nie.

3. Datagedrewe benadering

3.1 Inleiding tot masjienleer (ML)

ML-tegnieke implementeer gevorderde statistiese modelle om klassifikasie van onafhanklike veranderlikes te kan hanteer (StatSoft, 2004). Hierdie modelle sluit onder andere die volgende in: versteekte Markovmodelle (Bikel *et al.*, 1997), besluitnemingsboom-modelle (Sekine *et al.*, 1998), genetiese algoritmes (Srinivas & Patnaik, 1994), *k*-Naastebuurpuntmodelle (Daelemans *et al.*, 2003), neurale netwerke (Friedman & Kandel, 1999), ensovoorts.

Met betrekking tot mensetaaltegnologie is die gebruik van ML-tegnieke gewild in die ontwikkeling van sowel spraak- as tekstegnologie. Versteekte Markovmodelle word byvoorbeeld al vir baie jare met groot sukses in spraakprosessering aangewend (Lee, 1989; Rigoll, 1994; Tokuda *et al.*, 1995), terwyl verskeie ML-tegnieke gewild is in onder andere die inligtingonttrekkinggemeenskap (Bikel *et al.*, 1997; Collins & Singer, 1999).

Die doel van ML is om ’n rekenaarsisteem te leer om ’n bepaalde probleem op te los deur van vorige ondervinding gebruik te maak (Alpaydin, 2004), of om kennis uit voorafgeprosesseerde data te verwerf (Coiera, 1997) en dan hierdie kennis in te span wanneer dit ’n voorheen ongesiene probleem teëkom. Breedweg gesproke kan gesê word dat ’n masjien (rekenaar) leer wanneer dit self die struktuur, program of data op so ’n manier verander dat die verwagte toekomstige werkverrigting daarvan verbeter (Mitchell, 1997). Modules wat met behulp van ML ontwikkel word, leer gewoonlik uit groot hoeveelhede geannoteerde afrigtingsdata. Die uiteindelike

prestasie van die klassifiseerder hang daarom dikwels af van die kwaliteit en grootte van die afrigtingsdata: hoe meer data daar is, en hoe noukeuriger die data geannoteer is, hoe meer akkuraat is die uiteindelike klassifiseerder (Banko & Brill, 2001:30).

Datagedrewe modules, spesifiek modules wat met ML ontwikkel word, vereis daarom aanvanklik heelwat tyd vir die handmatige annotasie van afrigtingsdata. Deur middel van skoenlussteekproefneming (*bootstrapping*) word dit egter moontlik om hierdie annotasie semi-outomaties te doen, en die tyd wat nodig is om data te annoteer, neem dan vinnig af.

Uiteindelik moet die geannoteerde data omgeskakel word in afrigtingsdata voordat 'n ML-module daarmee afgerig kan word (vgl. 3.2.1). Voordat afrigting egter kan plaasvind, moet die ontwikkelaar besluit watter ML-algoritme vir dié doel gebruik gaan word. In die geval van die Afrikaanse woordafbreker en kompositumanaliseerder is besluit om dit met behulp van die Tilburg Memory-Based Learner te ontwikkel, wat kortliks in die volgende afdeling beskryf sal word. Daarna sal die ontwikkeling van groot hoeveelhede noukeurig geannoteerde afrigtingsdata in 3.1.2 beskou word.

3.1.1 Algoritme

Die Tilburg Memory-Based Learner (TiMBL; Daelemans *et al.*, 2003) is 'n geheuegebaseerde leerder waarin 'n verskeidenheid leermetodes (hoofsaaklik die *k*-Naastebuurpuntalgoritme) gebruik word om klassifiseerders te skep. TiMBL stoor 'n voorstelling van die afrigtingsdatastel eksplisiet in die geheue en klassifiseer dan nuwe gevalle op die basis van soortgelyke gevalle (d.i. *k*-Naastebuurpunte). Voordat klassifikasie van nuwe gevalle plaasvind, ken die leerder 'n gewig aan elke eienskap toe om die belangrikheid daarvan vir die leerproses te merk (Daelemans & Van Den Bosch, 2005). Eienskappe met hoër waardes word as belangriker in die klassifikasiefase beskou as dié met laer waardes.

TiMBL beskik oor verskeie parameters wat verstel kan word om die klassifiseerder te verbeter. Hierdie veelvuldige parameterinstellings maak van TiMBL 'n kragtige eksperimenteringsomgewing, waar die eindgebruiker byna volledige beheer het oor die aard, invoer en afvoer van die eksperimente. Vier van TiMBL se parameters is in hierdie studie ingespan om die klassifiseerder te ontwikkel, naamlik die tipe algoritme, afstandberekening, eienskapsgewigmoontlikhede en die hoeveelheid naastebuurpunte. (Vir teoretiese besonderhede rakende elk van hierdie parameters, vgl. Daelemans *et al.,* 2003.)

Die verandering van hierdie parameterinstellings kan die akkuraatheid van die klassifiseerder egter aansienlik beïnvloed (Daelemans *et al.*, 2003). Een metode om die beste parameterinstellings te bepaal, is om ’n ekstensiewe soektog van al die moontlike geldige permutasies van parameterinstellings uit te voer. Hierdie benadering is egter nie wenslik nie, aangesien dit bewerkingsintensief en tydrowend is om met al die permutasies op ’n volledige datastel te eksperimenteer. As alternatief kan begrensde progressiewe steekproefneming (*wrapped progressive sampling*; vgl. Van den Bosch, 2004) gebruik word om die beste kombinasies van algoritmiese parameters te voorspel. In al die eksperimente vir hierdie studie is *PSearch* (Groenewald, 2006) gebruik om die beste parameterkombinasies te bepaal.

3.1.2 Data

Aangesien daar by die aanvang van hierdie projek nie geannoteerde data vir die ontwikkeling van Afrikaanse kerntegnologieë beskikbaar was nie, moes afrigtingsdata aanvanklik handmatig geannoteer word. Handmatige annotasie is egter ook meestal ’n duur en tydrowende proses, maar anders as in die geval van reëlgebaseerde benaderings, kan moedertaalsprekers (wat nie noodwendig taaleksperts is nie) meestal ingespan word om hierdie annotasie te doen. Ongelukkig kan dit lei tot inkonsekwente annotasie en selfs tot foute, wat ’n invloed op die akkuraatheid van die klassifiseerder wat met hierdie data afgerig is, kan hê.

Om die akkuraatheid en spoed van die annotasie te verhoog, word *TurboAnnotate* gebruik (Van Huyssteen & Puttkammer, 2007). *TurboAnnotate* is ’n gebruikersvriendelike annoteringsomgewing wat vir die annotasie van taalkundige data vir ML-doeleindes ontwikkel is, en wat gebruik maak van skoenlussteekproefneming (*bootstrapping*). Met behulp van *TurboAnnotate* is dit moontlik om relatief vinnig akkuraat geannoteerde afrigtingsdata te genereer wat gebruik kan word om ’n woordafbreker en ’n kompositumanaliseerder vir Afrikaans te ontwikkel (sien Van Huyssteen & Puttkammer, 2007).

Handmatige kwaliteitskontrole is op steekproewe van die data gedoen om sodoende die kwaliteit van die annotasie te verseker. Nadat die data geannoteer is, is dit eers in afrigtingsdata omgesit, en daarna is die ML-algoritme daarmee afgerig. Hierdie twee prosesse word vervolgens onderskeidelik vir die woordafbreker en die kompositumanaliseerder in meer detail beskryf.

3.2 'n ML-woordafbreker

'n Aantal verskillende ML-tegnieke is al in die ontwikkeling van woordafbrekers gebruik. Daelemans en Van den Bosch (1992) gebruik byvoorbeeld truverbreidingsleer (*backpropagation learning*) in die ontwikkeling van 'n woordafbreker vir Nederlands. Wanneer naverwerking (*post processing*) by die sisteem geïnkorporeer word, is dit in staat om net minder as 96% van woordafbreekplekke korrek te identifiseer. Daelemans en Van den Bosch (1992) gee 'n kort oorsig oor sisteme wat met soortgelyke metodes ontwikkel is, waaronder sisteme wat met behulp van truverbreiding ontwikkel is (Plunkett & Marchman, 1989) en ook konneksionistiese leer (Rumelhart & McClelland, 1986; Fritzke & Nasahl, 1991).

Fick (2003) beskryf die ontwikkeling van 'n Afrikaanse woordafbreker wat gebruik maak van 'n neurale netwerk. Die neurale netwerk is afgerig met 52 167 woorde met lettergreepaanduidings, wat uit die elektroniese weergawe van die *Handwoordeboek van die Afrikaanse Taal* (Odendaal *et al.,* 1983) onttrek is. Die klassifiseerder het in verskillende evaluasies tussen 97,56% en 98,75% woordposisies korrek as óf geldige, óf ongeldige afbrekingspunte geklassifiseer. Fick (2003) toon daarom duidelik aan dat neurale netwerke geskik is vir die ontwikkeling van 'n Afrikaanse woordafbreker. Dit is egter nog nie duidelik of ander ML-algoritmes dalk meer geskik is nie, en daarom word hier met 'n ander algoritme (TiMBL) geëksperimenteer.

3.2.1 Data

Die ML-woordafbreker is afgerig met 39 943 woorde – ongeveer 35% minder woorde as wat Fick (2003) gebruik het – waarin elke moontlike afbreekplek met 'n asterisk (*) aangedui is. Die woord *fakulteitsraad* word dus in die geannoteerde data as *fa*kul*teits*raad* voorgestel. Die afrigtingsdata bevat 124 595 afbreekplekke, wat beteken daar is gemiddeld 3,67 afbreekplekke per woord. Die gemiddelde woordlengte is 12,78 karakters per woord (afbreekplekke uitgesluit), en die data bevat 927 negatiewe afrigtingsinstansies (woorde wat nie afgebreek kan word nie).

Hierdie geannoteerde data is egter nie direk bruikbaar nie, aangesien ML-algoritmes normaalweg vasgestelde lengte-eienskapsvektore as toevoer vereis. Die data moet daarom in afrigtingsdata wat vir die ML-algoritme geskik is, omgesit word.

Die doel hier is om 'n klassifiseerder te ontwikkel wat vir elke posisie in 'n woord kan aandui of dit 'n afbreekplek in die woord is, al dan

nie. Die klassifiseerder het ’n aantal eienskappe rondom ’n potensiële afbreekplek nodig om te leer binne watter konteks ’n afbreekplek tussen twee betrokke letters ingevoeg moet word. ’n Aantal karakters word dus aan weerskante van die betrokke posisie binne ’n woord in ag geneem tydens die besluitnemingsproses. Dit impliseer dat ’n skuiwende venster van ’n voorafbepaalde grootte vir elke posisie in die woord gebruik word. Die lengte-eienskapsvektore wat uit die geannoteerde data gegenereer is, word in die volgende afdeling bespreek.

3.2.2 Eienskappe

Tabel 1 toon hoe die woord *fakulteitsraad* in die afrigtingsdata voorgestel word. Daaruit blyk dit dat dié woord veertien verskillende afrigtingsgevalle (of lengte-eienskapsvektore) tot die afrigtingsdata bydra.

Tabel 1: Voorstelling van *fakulteitsraad* in die afrigtingsdata van die ML-woordafbreker

⬇						
Linkskonteks			**Regskonteks**			**Klas**
_	_	_	f	a	k	=
_	_	f	a	k	u	=
_	f	a	k	u	l	*
f	a	k	u	l	t	=
a	k	u	l	t	e	=
k	u	l	t	e	i	*
u	l	t	e	i	t	=
l	t	e	i	t	s	=
t	e	i	t	s	r	=
e	i	t	s	r	a	=
i	t	s	r	a	a	*
t	s	r	a	a	d	=
s	r	a	a	d	_	=
r	a	a	d	_	_	=
a	a	d	_	_	_	=

In **Error! Reference source not found.** word 'n konteks van drie karakters links en drie karakters regs van die potensiële afbreekplek gebruik. Hierdie ses karakters is die eienskappe van die betrokke afrigtingsgeval. Die linkerkonteks is aanvanklik leeg, terwyl die eerste drie karakters van die woord in die regterkonteks opgeneem is. 'n Klas vir die afbreekplek word in die laaste kolom gegee – in die eerste geval is hierdie klas nie 'n asterisk nie, aangesien daar nie 'n afbreekplek op die posisie (d.i. voor die eerste letter van die woord) ter sprake is nie. 'n Gelykaanteken (=) word toegeken aan 'n posisie wat nie 'n afbreekplek in die betrokke woord is nie.

In die volgende stap skuif die venster een karakter na regs, en die eerste karakter van die woord is nou in die eerste kontekssspasie van die linkerkonteks. Die volgende karakter van die woord word nou in die laaste regterkontekssspasie opgeneem, en 'n klas word aan die nuwe potensiële afbreekplek toegeken. Die proses word herhaal totdat al drie die spasies van die regterkonteks leeg is en die hele woord ondersoek is. Uit die klastoekenning blyk dit dat daar drie afbreekplekke in die woord voorkom, aangesien daar drie posisies is waarvan die klas 'n asterisk is (*fa*kul*teits*raad*).

Om die eienskapseleksie te vergroot, kan die kontekste na links en regs vergroot word. Deur die konteks byvoorbeeld na vier te vergroot, bevat elke afrigtingsgeval agt eienskappe. Die grootte van die konteks (die hoeveelheid eienskappe) speel 'n belangrike rol in die akkuraatheid van die klassifiseerder, aangesien die klassifiseerder 'n sekere hoeveelheid karakters in ag neem tydens die klassifikasie van 'n potensiële afbreekplek. 'n Konteks wat te klein is, verskaf nie genoegsame inligting nie, terwyl 'n te groot konteks weer veroorsaak dat die klassifiseerder te spesifiek afgerig is en dus nie nuwe, ongesiene gevalle reg kan klassifiseer nie. Tydens die eksperimentele ontwikkelingsfase moet die optimale grootte van die konteks dus ook bepaal word.

3.2.3 Evaluasie

Deur gebruik te maak van *PSearch* is bepaal dat 'n linker- en regterkonteks van ses, die algemene *k*-Naastebuurpuntalgoritme (IB1), oorvleuelingsmetriek (O) as afstandsberekening, 'n eienskapsgewigmoontlikheid van inligtingswinsgewigstoekenning (IG) en een naastebuurpunt die beste resultate lewer wanneer 'n ML-woordafbreker vir Afrikaans afgerig word. 'n Klassifiseerder met hierdie parameterinstellings is vervolgens afgerig en met behulp van tienvoudige kruisvalidasie geëvalueer (op dieselfde vlakke soos bespreek in 2.1). Die ML-woordafbreker behaal 'n *f*-telling van 98,11% (met pre-

sisie van 98,21% en herroeping van 98,00%) op afbreekplekke, en op woordvlak behaal die woordafbreker ’n akkuraatheid van 91,94%.

Die resultate wat met die TiMBL-klassifiseerder verkry is, vergelyk goed met dié van Fick (2003), aangesien dit onderskeidelik 0,55% beter en 0,54% swakker vaar as die neurale netwerk. Die feit dat die TiMBL-klassifiseerder dieselfde resultate as die neurale netwerk lewer met heelwat minder afrigtingsdata (d.i. ongeveer 35% minder), laat ons tot die gevolgtrekking kom dat geheuegebaseerde leer wel meer effektief is vir die ontwikkeling van ’n Afrikaanse woordafbreker as neurale netwerke.

3.3 ’n ML-kompositumanaliseerder

Uit die literatuur blyk dat daar nog bykans geen navorsing oor die effektiwiteit van ML vir die ontwikkeling van ’n kompositumanaliseerder gedoen is nie. Witschel en Biemann (2005) maak van kompakte Patricia-bome (*compact Patricia tries*) gebruik in die ontwikkeling van ’n kompositumanaliseerder vir Duits. Hulle eksperimenteer met verskillende hoeveelhede eienskappe (*features*), en eers wanneer die sisteem 1 000 of meer eienskappe gebruik, behaal dit ’n presisie hoër as 80%. Ten spyte van hierdie ontmoedigende resultate en by gebrek aan vorige navorsing oor die onderwerp, is tog besluit om ’n kompositumanaliseerder met behulp van ML te ontwikkel.

3.3.1 Data

Die ML-kompositumanaliseerder is afgerig met 77 589 woorde wat met twee verskillende simbole geannoteer is: ’n onderstreep (_) om valensiemorfeme aan te dui en ’n plus (+) om woordgrense aan te dui. Die woord *hondehok* word dus geannoteer as *hond _ e + hok* en *fakulteitsraad* as *fakulteit _ s + raad.* Die data bevat 101 092 morfeemgrense, wat beteken dat daar 1,30 posisies per woord is wat met een van die bogenoemde simbole geannoteer is. Die data bevat ook 4 846 negatiewe afrigtingsinstansies (woorde wat nie komposita is nie en dus nie deur die kompositumanaliseerder geanaliseer behoort te word nie).

Dit is belangrik om te noem dat die manier waarop die woorde geanaliseer word niks impliseer oor die volgorde van kombinasie nie. Die feit dat ’n onderstreep gebruik word voor ’n valensiemorfeem, impliseer nie dat die valensiemorfeem eerste met die linkerkantste konstituent verbind nie. Die simbole is slegs so gekies om te kan onderskei tussen onafhanklike konstituente, wat deur ’n plus (+)

voorafgegaan word, en valensiemorfeme, wat deur 'n onderstreep (_) voorafgegaan word.

3.3.2 Eienskappe

Die woorde is op dieselfde manier as dié van die ML-woordafbreker in afrigtingsdata omgesit, met die verskil dat die klas van 'n betrokke posisie in 'n woord nou die volgende kan wees: 'n gelykaanteken vir 'n posisie wat nie 'n onafhanklikekonstituentgrens of valensiemorfeemgrens kan wees nie; 'n onderstreep vir valensiemorfeemgrense; of 'n plus vir onafhanklikekonstituentgrense. In Tabel 2 word aangetoon hoe die woord *fakulteitsraad* in afrigtingsdata vir die ML-kompositumanaliseerder omgesit is. In hierdie geval word weereens 'n konteks van drie karakters voor en drie karakters na die posisie ter sprake gebruik. Elke afrigtingsgeval het dus ook ses eienskappe.

Tabel 2: Voorstelling van *fakulteitsraad* in die afrigtingsdata van die ML-kompositum-analiseerder-afrigtingsdata

⬇						
Linkskonteks			**Regskonteks**			**Klas**
_	_	_	f	a	k	=
_	_	f	a	k	u	=
_	f	a	k	u	l	=
f	a	k	u	l	t	=
a	k	u	l	t	e	=
k	u	l	t	e	i	=
u	l	t	e	i	t	=
l	t	e	i	t	s	=
t	e	i	t	s	r	=
e	i	t	s	r	a	_
i	t	s	r	a	a	+
t	s	r	a	a	d	=
s	r	a	a	d	_	=
r	a	a	d	_	_	=
a	a	d	_	_	_	=

Uit Tabel 2 blyk dat daar twee morfeemgrense in die woord *fakulteitsraad* voorkom: een wat aandui dat die daaropvolgende morfeem 'n valensiemorfeem is (tussen *-eit* en *sra-*), en een wat aandui dat die daaropvolgende morfeem 'n onafhanklike konstituent is (tussen *-its* en *raa-*). Hierdie afrigtingsdata word vervolgens gebruik om 'n TiMBL-klassifiseerder af te rig.

3.3.3 Evaluasie

Met behulp van *Psearch* is bepaal dat 'n konteks van agt karakters weerskante van die punt ter sprake, die algemene *k*-Naastebuurpuntalgoritme (IB1), oorvleuelingsmetriek (O) as afstandsberekening, 'n eienskapsgewigmoontlikheid van inligtingswinsgewigstoekenning (IG) en sewe naastebuurpunte die beste resultate lewer wanneer 'n kompositumanaliseerder vir Afrikaans ontwikkel word. 'n Klassifiseerder met hierdie parameterinstellings is afgerig en met behulp van tienvoudige kruisvalidasie geëvalueer (op dieselfde vlakke soos bespreek in 2.2 hierbo). Die ML-kompositumanaliseerder behaal 'n *f*-telling van 90,57% (met 93,74% presisie en 87,60% herroeping) op konstituentgrense en 'n akkuraatheid van 81,28% op woordvlak.

Dit is interessant om te merk dat die ML-kompositumanaliseerder steeds heelwat swakker vaar as die ML-woordafbreker, ten spyte van die feit dat amper dubbeld soveel afrigtingsdata gebruik is. Die kompositumanalisetaak is dus meer kompleks as wat aanvanklik geantisipeer is, en al lewer die ML-kompositumanaliseerder beter resultate as die LSP-analiseerder, moet dit waarskynlik verder verbeter word voordat dit as deel van 'n morfologiese analiseerder geïmplementeer kan word. Die analiseerder wat hier ontwikkel is, kan egter in die toekoms met vrug gebruik word om addisionele afrigtingsdata semi-outomaties te genereer. Sodoende sou 'n meer akkurate kompositumanaliseerder ontwikkel kon word.

4. Slot

In hierdie artikel is die ontwikkeling van 'n woordafbreker en kompositumanaliseerder vir Afrikaans beskryf. Daar is eers op die ontwikkeling van reëlgebaseerde weergawes van hierdie kerntegnologieë gefokus, maar dit het geblyk dat die tegnieke wat gebruik is, tekort geskiet het. Vervolgens is besluit om eerder van ML gebruik te maak om 'n woordafbreker en kompositumanaliseerder vir Afrikaans te ontwikkel. Die resultate van die vier ontwikkelde modules word in Tabel 3 opgesom. Aangesien sowel die ML-woordafbreker as die ML-kompositumanaliseerder bevredigende resultate lewer,

wil dit voorkom asof die datagedrewe benadering geskik is vir die ontwikkeling van kerntegnologieë vir Afrikaans.

Tabel 3: Opsomming van resultate van ontwikkelde kern-tegnologieë

		Presisie	Herroeping	f-telling	Akkuraatheid
Woord-afbreker	Reëlgebaseerd	92,03	89,69	90,84	73,56
	Masjienleer	98,21	98,00	98,11	91,94
Kompositum analiseerder	Reëlgebaseerd	96,84	65,57	78,20	66,40
	Masjienleer	93,74	87,60	90,57	81,28

ML is ook intussen gebruik om 'n lemma-identifiseerder vir Afrikaans te ontwikkel (Groenewald, 2006). Hierdie lemma-identifiseerder is afgerig met 72 226 woorde, en nadat die parameters van die ML-algoritme geoptimaliseer is, bereik dit 'n akkuraatheid van 92,80%. ML sal ook in die toekoms gebruik word om ander kerntegnologieë, soos 'n sintaktiese analiseerder, vir Afrikaans te ontwikkel.

Geraadpleegde bronne

ALPAYDIN, E. 2004. Introduction to machine learning. Cambridge: MIT.

BANKO, M. & BRILL, E. 2001. Scaling to very large corpora for natural language disambiguation. Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics and the 10th Conference of the European Chapter of the Association for Computational Linguistics, Toulouse. p. 26-33. http://acl.ldc.upenn.edu//P/P01/P01-1005.pdf Date of access: 31 Oct. 2005.

BIKEL, D., MILLER, S., SCHWARTZ, R. & WEISCHEDEL, R. 1997. Nymble: a high-performance learning name-finder. Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, D.C. p. 194-201. http://arxiv.org/PS_cache/cmp-lg/pdf/9803/ 9803003.pdf Date of access: 31 Oct. 2005.

BOOT, M. 1984. Taal, tekst, computer. Katwijk: Servire.

COIERA, E. 1997. Guide to medical informatics, the internet and telemedicine. http://www.uhnresearch.ca/centres/ehealth/html/glossary/eh_glossary.shtml Date of access: 15 Jan. 2005.

COLLINS, M. & SINGER, Y. 1999. Unsupervised models for named entity classification. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Maryland, USA. p. 189-196. http://acl.ldc.upenn.edu/ W/W99/W99-0613.pdf Date of access: 31 Oct. 2005.

CTexT. 2005. Afrikaanse speltoetser 3.0. Potchefstroom: Noordwes-Universiteit.

DAELEMANS, W. 1989. Automatic hyphenation: linguistics versus engineering. (*In* Heyvaert, F.J. & Steurs, F., *eds*. Worlds behind words. Leuven: Leuven University Press. p. 347-364.)

DAELEMANS, W., BINNENPOORTE, D., DE VRIEND, F., STURM, J., STRIK, H. & CUCCHIARINI, C. 2005. Establishing priorities in the development of HLT resources: the Dutch-Flemish experience. (*In* Daelemans, W., Du Plessis, T., Snyman, C. & Teck, L., *eds*. Multilingualism and electronic language management. Pretoria: Van Schaik. p. 9-23.)

DAELEMANS, W. & VAN DEN BOSCH, A. 1992. Generalisation performance of backpropagation learning on a syllabification task. (*In* Drossaers, M. & Nijholt, A., *eds*. TWLT3: connectionism and natural language processing. Enschede: Twente University. p. 27-38.)

DAELEMANS, W. & VAN DEN BOSCH, A. 2005. Memory-based language processing: studies in natural language processing. Cambridge: Cambridge University Press.

DAELEMANS, W., ZAVREL, J., VAN DER SLOOT, K. & VAN DEN BOSCH, A. 2003. Timbl: Tilburg memory based learner, version 5.0: reference guide. Tilburg: Tilburg University. (Technical Report, ILK 03-10.) http://ilk.uvt.nl/downloads/pub/papers/ ilk0310.ps.gz Date of access: 31 Oct. 2005.

FICK, M. 2003. Neurale netwerke as moontlike woordafkappingstegniek vir Afrikaans. *Suid-Afrikaanse tydskrif vir natuurwetenskap en tegnologie*, 22(1):2-5.

FRIEDMAN, M. & KANDEL, A. 1999. Introduction to pattern recognition: statistical, structural, neural, and fuzzy logic approaches. River Edge: World Scientific.

FRITZKE, B. & NASAHL, C. 1991. A neural network that learns to do hyphenation. (*In* Kohonen, T., Makisara, K., Simula, O. & Kangas, J., *eds*. Artificial neural networks. Proceedings of the International Conference on Artificial Neural Networks. Amsterdam: Elsevier Science Publishers. p. 1375-1378.)

GROENEWALD, H.J. 2006. Automatic lemmatisation for Afrikaans. Potchefstroom: Noordwes-Universiteit. (Unpublished dissertation.)

JURAFSKY, D. & MARTIN, J.H. 2000. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River: Prentice Hall.

KOSKENNIEMI, K. 1983. Two-level morphology: a general computational model for word-form recognition and production. Helsinki: University of Helsinki. (Ph.D. thesis.)

LEE, K. 1989. Automatic speech recognition: the development of the Sphinx system. Boston: Kluwer Academic Publishers.

LEZIUS, W., RAPP, R. & WETTLER, M. 1996. A morphology-system and part-of-speech tagger for German. (In Gibbon, D., ed. Natural language processing and speech technology. Berlin: De Gruyter. p. 369-378.)

LIANG, M. 1983. Word hy-phen-a-tion by Com-put-er. Stanford: Stanford University. (Ph.D. thesis.)

MINNEN, G., CARROLL, J. & PEARCE, D. 2001. Applied morphological processing of English. *Natural language engineering*, 7(3): 207-223.

MITCHELL, T.M. 1997. Machine learning. New York: McGraw-Hill.

NUNN, A. 1999. Automatic hyphenation of Dutch words based on linguistic rules. Computational Linguistics in the Netherlands 1999: Selected Papers from the 10th CLIN Meeting, Utrecht. http://wwwuilots.let.uu.nl/publications/clin1999/Pap/nunn.pdf Date of access: 29 Feb. 2008.

ODENDAAL, F.F., SCHOONEES, P.C., SWANEPOEL, C.J., DU TOIT, S.J. & BOOYSEN, C.M. 1983. Verklarende Handwoordeboek van die Afrikaanse Taal. Johannesburg: Perskor.

PLUNKETT, K. & MARCHMAN, V. 1989. Pattern association in a back propagation network: implications for language acquisition. San Diego: UCSD Center for Research in Language. (Technical Report, 8902.)

RIGOLL, G. 1994. Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems. *Speech and audio processing, IEEE Transactions*, 2(1):175-184.

RUMELHART, D.E. & MCCLELLAND, J. 1986. On learning the past tense of English verbs. (*In* Rumelhart, D.E., McCleland, J.L & the PDP Research Group, eds. Parallel distributed processing: explorations in the microstructure of cognition. Vol. 2. Cambridge: Bradford.

SCHILLER, A. 2005. German compound analysis with wfsc. Proceedings of the 5th International Workshop of Finite State Methods in Natural Language Processing (FSMNLP 2005), Helsinki. p. 239-246.

SEKINE, S., GRISHMAN, R. & SHINNOU, H. 1998. A decision tree method for finding and classifying names in Japanese texts. Proceedings of 6th Workshop on Very Large Corpora. http://acl.ldc.upenn.edu/W/W98/W98-1120.pdf Date of access: 31 Oct. 2005.

SPROAT, R. 1992. Morphology and computation. Cambridge: MIT.

SRINIVAS, M. & PATNAIK, L.M. 1994. Genetic algorithms: a survey. *IEEE computer*, 27(6):17-26.

STATSOFT, INC. 2004. Electronic statistics textbook. Tulsa: StatSoft. http://www.statsoft.com/textbook/stathome.html. Date of access: 31 Oct. 2005.

SUID-AFRIKAANSE AKADEMIE VIR WETENSKAP EN KUNS. 2002. Afrikaanse woordelys en spelreëls. 9e dr. Kaapstad: Pharos.

TEXT ANALYSIS INTERNATIONAL. 2001. Integrated development environments for natural language processing. http://www. textanalysis.com Date of access: 15 Jun. 2007.

TOKUDA, K., KOBAYASHI, T. & IMAI, S. 1995. Speech parameter generation from HMM using dynamic features. Acoustics, speech, and signal processing. *ICASSP-95*, 1(1):660-663.

TUTELAERS, P. 1993. Herziene afbreekpatronen voor het Nederlands. *MAPS*, 1993:187-190.

VAN DEN BOSCH, A. 2004. Paramsearch 1.0 Beta Patch 24. http://ilk.uvt.nl/software.html/paramsearch Date of access: 20 March 2007.

VAN DEN BOSCH, A. & DAELEMANS, W. 1999. Memory-based morphological analysis. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL'99, University of Maryland, USA, June 20-26. p. 285-292.

VAN HUYSSTEEN, G.B. & PUTTKAMMER, M.J. 2007. Accelerating the annotation of lexical data for less-resourced languages. Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007), Antwerp, Belgium. p. 1505-1508.

VAN HUYSSTEEN, G.B. & VAN ZAANEN, M.M. 2003. A spellchecker for Afrikaans, based on morphological analysis. (*In* De Schryver, G., *ed*. 6th International Terminology in Advanced Management Applications Conference, Conference Proceedings. Pretora: $(SF)^2$ Press. p. 189-194.)

VAN RIJSBERGEN, C.J. 1979. Information retrieval. 2nd ed. London: Butterworths.

VANDEGHINSTE, V. 2002. Automated compounding as a means for maximizing lexical coverage in speech recognition. (*In* Theune, M., Nijholt, A. & Hondorp, H., *eds*. Computational linguistics in the Netherlands 2001: selected papers from the 12th CLIN meeting, Amsterdam. p. 190-203.)

VOUTILAINEN, A. 1999. A short history of tagging. (*In* Van Halteren, H., *ed*. Syntactic wordclass tagging. Dordrecht: Kluwer. p. 9-21.)

WITSCHEL, F. & BIEMANN, C. 2005. Rigorous dimensionality reduction through linguistically motivated feature selection for text categorisation. Proceedings of NODALIDA 2005, Joensuu. http://phon.joensuu.fi/lingjoy/01/witschelF.pdf Date of access: 2 Feb. 2008.

Kernbegrippe:

Afrikaanse taalkunde
kerntegnologieë
kompositumanaliseerder
masjienleer
woordafbreker

Key concepts:

Afrikaans linguistics
compound analyser
core-technologies
hyphenator
machine learning