Die ontwikkeling van ’n fleksievorm-genereerder vir Afrikaans

S. Pilon
Sentrum vir Tekstegnologie (CTexT)
Potchefstroomkampus
Noordwes-Universiteit
POTCHEFSTROOM
E-pos: Sulene.Pilon@nwu.ac.za

Abstract

The development of an inflected form generator for Afrikaans

In this article the development of an inflected form generator for Afrikaans is described. Two requirements are set for this inflected form generator, viz. to generate only one specific inflected form of a lemma and to generate all possible inflected forms of a lemma. The decision to use machine learning instead of the more traditional rule-based approach in the development of this core-technology is explained and a brief overview of the development of LIA, a lemmatiser for Afrikaans, is given. Experiments are done with three different methods and it is shown that the most effective way of developing an inflected form generator for Afrikaans is by training different classifiers for each affix. Therefore a classifier is trained to generate a plural form, one to generate the diminutive, one to generate the plural of diminutive, et cetera. The final inflected form generator for Afrikaans (AIL-3) reaches an average accuracy of 86,37% on the training data and 86,88% on a small amount of new data. It is indicated that, with the help of a pre-processing module, AIL-3 meets the requirements that were set for an Afrikaans inflected form generator. Finally suggestions are made on how to improve the accuracy of AIL-3.

Opsomming

Die ontwikkeling van ’n fleksievormgenereerder vir Afrikaans

In hierdie artikel word die ontwikkeling van ’n fleksievorm-genereerder vir Afrikaans beskryf. Twee vereistes waaraan hierdie fleksievormgenereerder moet voldoen, word gestel, te

wete om slegs een spesifieke fleksievorm van ’n lemma te kan genereer en om alle moontlike fleksievorme van ’n lemma te kan genereer. Die besluit om masjienleertegnieke te gebruik in die ontwikkeling van hierdie kerntegnologie in plaas van reëlgebaseerde metodes, wat die tradisionele benadering in die ontwikkeling van fleksievormgenereerders is, word verduidelik en ’n kort oorsig oor die ontwikkeling van LIA, ’n lemma-identifiseerder vir Afrikaans, word gegee. Daarna word geëksperimenteer met drie verskillende ontwikkelingsmetodes en uiteindelik word die fleksievormgenereerder ontwikkel deur verskillende klassifiseerders vir elke moontlike fleksievorm af te rig; dit wil sê daar is uiteindelik ’n klassifiseerder wat meervoudsvorme genereer, een wat diminutief genereer, een wat die meervoud van die diminutief genereer, een wat die attributiewe vorm van adjektiewe genereer, ensovoorts. Hierdie fleksievormgenereerder (AIL-3) bereik ’n gemiddelde akkuraatheid van 86,37% op die afrigtingsdata en 86,88% op ’n klein hoeveelheid nuwe data. Daar word aangetoon dat ’n voorverwerkingsmodule tot AIL-3 toegevoeg kan word om te verseker dat dit voldoen aan die vereistes wat vir ’n fleksievormgenereerder vir Afrikaans gestel is en uiteindelik word voorstelle gemaak oor hoe om die akkuraatheid van hierdie fleksievormgenereerder verder te verbeter.

1. Inleiding

Die Sentrum vir Tekstegnologie (CTexT) in die Navorsingseenheid Taal en Literatuur in die Suid-Afrikaanse Konteks aan die Noordwes-Universiteit (Potchefstroomkampus) is tans besig met die ontwikkeling van ’n masjienvertalingsisteem (MV-sisteem) met Engels as brontaal en Afrikaans as teikentaal. Aangesien daar nie genoeg parallelle Engels-Afrikaans-korpora bestaan nie, word die METIS II-benadering (http://www.ilsp.gr/metis2/; vgl. ook Dirix *et al.*, 2005) gebruik in die ontwikkeling van die MV-sisteem. Binne hierdie benadering word woorde in die brontaalteks gelemmatiseer en woord vir woord met behulp van ’n tweetalige woordeboek na die teikentaal vertaal. Daarna moet die teikentaallemmas na hulle oorspronklike vorm (d.i. soos dit in die brontaaldokument voorgekom het) herlei word. Byvoorbeeld: In die sin *He will meet his friends and colleagues after the ceremony* sal die woorde *friends* en *colleagues* in die eerste stap gelemmatiseer word tot *friend<<PL>>* en *colleague<<PL>>* wat dan met behulp van ’n tweetalige woordeboek vertaal word na *vriend<<PL>>* en *kollega<<PL>>*. Hierdie vorms moet dan deur ’n “omgekeerde lemmatiseerder” (fleksievormgenereerder) verander word in *vriende* en *kollegas.*

Verder word ’n omvattende leksikale databasis vir Afrikaans, ALEXANDER (*Afrikaans lexicon and annotated database for engineering and research*), ook deur CTexT ontwikkel. Hierdie databasis sal inligting oor Afrikaanse woordvorms bevat, waaronder byvoorbeeld lemma-inligting, volledige morfologiese analises, Engelse vertaalekwivalente, fonetiese transkripsies, skakels na die Afrikaanse woordnet, ensovoorts

Een van die velde wat by lemmas gevul moet word, is alle moontlike fleksievorms van daardie lemma. Dit is binne hierdie konteks dus belangrik om te weet dat die lemma *hond* die fleksievorms *honde, hondjie* en *hondjies* kan hê. Om al hierdie fleksievorms uit korpora te onttrek, is tydrowend en dikwels onvolledig – daarom moet alternatiewe maniere gevind word om fleksievorms van lemmas outomaties te genereer.

Vir doeleindes van die MV-sisteem en ALEXANDER wat tans onder ontwikkeling is, moet dus ’n fleksievormgenereerder ontwikkel word wat

- slegs een spesifieke fleksievorm van ’n lemma kan genereer; en
- alle fleksievorms van ’n bepaalde lemma kan genereer.

Die ontwikkeling van ’n fleksievormgenereerder wat aan hierdie vereistes voldoen, sal in hierdie artikel beskryf word.

In die volgende afdeling sal die tradisionele benadering tot morfologiese analise en generering, tweevlakmorfologie, bespreek word waarna die Afrikaanse lemma-identifiseerder (LIA) kortliks beskryf sal word. Daar sal aangetoon word dat die tradisionele benadering nie die mees geskikte is vir ’n Afrikaanse lemma-identifiseerder nie en ’n alternatiewe ontwikkelingsmetode, naamlik masjienleer, sal voorgestel word. Die rede vir die keuse van die ontwikkelingsmetode sal verduidelik word en daarna sal die ontwikkeling van die Afrikaanse fleksievormgenereerder beskryf word.

2. Tweevlakmorfologie

Tweevlakmorfologie (*two-level morphology*) is vir die eerste keer in 1983 deur Koskenniemi voorgestel en sedertdien is dit gebruik om morfologiese analiseerders en genereerders vir verskeie tale te ontwikkel. Tweevlakmorfologie is gebaseer op die aanname dat dieselfde reëls wat gebruik word om morfologies komplekse woorde te herken en te analiseer, ook gebruik kan word om morfologies komplekse woorde te genereer (Koskenniemi, 1983; 1986).

Nadat Koskenniemi tweevlakmorfologie vir die eerste keer in 1983 geïmplementeer het, het verskeie ander implementasies spoedig gevolg. Ander implementasies sluit in dié van Karttunen (1983) en van Beesley (1989; 1990) wat spesifiek fokus op die morfologiese analise van Arabies. Ook in Europa het navorsers tweevlakmorfologie gebruik om morfologiese analiseerders te ontwikkel wat in verskeie groot MTT-sisteme geïmplementeer is (vgl. byvoorbeeld Black *et al.*, 1987; Ritchie *et al.*, 1987; 1992; Carter, 1995 en Armstrong, 1996).

Aangesien tweevlakmorfologie 'n geskikte benadering tot die ontwikkeling van morfologiese analiseerders (en dus ook morfologiese genereerders) blyk te wees, sou 'n Afrikaanse morfologiese analiseerder ook met behulp van hierdie metode ontwikkel kon word. Vir Afrikaans sou dit byvoorbeeld moontlik wees om 'n reël te skryf wat soos in die vereenvoudigde voorbeeld hieronder lyk.

1. Onskeibare werkwoord (teenwoordige tyd)⇔ Onskeibare werkwoord (verlede tyd)

2. ge + Onskeibare werkwoord (teenwoordige tyd) = Onskeibare werkwoord (verlede tyd)

Die eerste reël kan geïnterpreteer word as:

> Dit is moontlik om 'n onskeibare werkwoord in die teenwoordige tyd te verander na 'n onskeibare werkwoord in die verlede tyd. Dit is ook moontlik om 'n onskeibare werkwoord in die verlede tyd te verander na 'n onskeibare werkwoord in die teenwoordige tyd.

Op grond van die feit dat dit moontlik is om hierdie twee vorms te verander, is die tweede reël nodig om die een vorm in die ander te verander. Dit sê dat die verlede tyd van 'n onskeibare werkwoord (bv. *gesleep*) gelyk is aan die teenwoordige tyd van dieselfde onskeibare werkwoord met *ge-* vooraan (dus *ge* + *sleep*). As die teenwoordigetydsvorm na die verledetydsvorm verander moet word, moet *ge-* vooraan die werkwoord gevoeg word; as die verledetydsvorm na die teenwoordigetydsvorm verander moet word, moet die *ge-* vooraan die werkwoord verwyder word. Dit is met hierdie reëls dus moontlik om 'n werkwoord soos *geloop* te lemmatiseer na *loop* of om van die lemma *loop* die verledetydsvorm *geloop* te genereer.

Daar kan soortgelyke reëls geformuleer word vir al die fleksieprosesse in Afrikaans (meervoud, verkleining, deelwoorde, attribu-

tiewe *-e*, ensovoorts) en sodoende kan ’n omkeerbare lemma-identifiseerder ontwikkel word. Die probleem met hierdie benadering lê egter in die groot hoeveelheid uitsonderings wat altyd voorkom wanneer ’n reël vir ’n linguistiese fenomeen geformuleer moet word. Meervoude in Afrikaans word byvoorbeeld prototipies met een van twee morfeme, naamlik die *-s* en die *-e*, gevorm. Alhoewel dit moeilik is om voorwaardes te formuleer vir wanneer watter een van die morfeme gebruik moet word, kan ruweg gestel word dat naamwoorde met net een lettergreep (bv. *vurk, baas, bal* en *gas*) die morfeem *-e* (*vurke, base, balle* en *gasse/gaste*) neem wanneer die meervoud gevorm word, terwyl meerlettergrepige woorde (soos *tafel, lepel, waaier* en *rekenaar*) die *-s* neem (*tafels, lepels, waaiers* en *rekenaars*).

Uit die voorbeelde van naamwoorde wat *-e* neem, blyk dit dat daar met verskeie morfonologiese veranderings rekening gehou moet word wanneer die meervoud van eenlettergrepige naamwoorde gevorm word. Verder is daar ook talle uitsonderings op die reël (bv. *man x mans, arend x arende, hotel x hotelle/hotels*). Benewens hierdie uitsonderings veroorsaak komposita ’n verdere probleem. Wanneer twee eenlettergrepige woorde (bv. *vis* en *vurk*) met mekaar verbind, vorm dit ’n meerlettergrepige woord (*visvurk*) wat, volgens die reël wat hierbo gepostuleer is, ’n *-s* sal neem wanneer meervoud gevorm word. Die meervoud van komposita word egter gevorm op grond van die laaste konstituent daarvan en daarom is die korrekte meervoudsvorm *visvurke* en nie **visvurks* nie.

Dit is, met inagneming van net die twee prototipiese morfeme, reeds moeilik om reëls te formuleer vir meervoudsvorming in Afrikaans. Daar moet egter verder in ag geneem word dat daar etlike allomorwe vir albei hierdie morfeme bestaan. Voorbeelde van hierdie allomorwe sluit in *-ens* (*beddens*)*, -ers* (*kalwers*), *-ë* (*vlieë*), *-’s* (*ski’s*), *-a* (*sentra*), *-ci* (*politici*), ensovoorts.

Ten spyte van die problematiek wat die reëlgebaseerde benadering inhou, is by CTexT ’n reëlgebaseerde stam- en lemma-identifiseerder (RAGEL: *Reëlgebaseerde Afrikaanse grondwoord en lemma-identifiseerder*) ontwikkel, aangesien daar nie afrigtingsdata beskikbaar was waarmee ’n masjienleeralgoritme afgerig kon word nie. Daar is ongeveer twee jaar aan RAGEL gewerk. Geen formele evaluasie van hierdie kerntegnologie is gedoen nie, maar op 1 000 lukraakgeselekteerde morfologies komplekse woorde, het RAGEL ’n teleurstellende akkuraatheid van 67% behaal. Daar is besluit dat hierdie reëlgebaseerde metode waarskynlik nie die mees geskikte is om te gebruik wanneer ’n lemmatiseerder vir Afrikaans ontwikkel

word nie en daarom is 'n lemma-identifiseerder (LIA: Lemma-identifiseerder vir Afrikaans) ontwikkel met behulp van masjienleertegnieke.

3. 'n Lemma-identifiseerder vir Afrikaans

Groenewald (2006) ontwikkel LIA met behulp van TiMBL (Daelemans *et al.*, 2004), 'n geheuegebaseerdeleersisteem (*memory-based learning system*) wat 'n aantal doeltreffende algoritmes insluit en ten doel het om klassifiseerders vir 'n spesifieke taak te konstrueer. Elke algoritme het verskeie parameterinstellings wat verstel kan word om 'n klassifiseerder pas te maak vir 'n spesifieke taak. Vir 'n volledige beskrywing van die verskillende algoritmes en parameterinstellings, kyk Daelemans *et al.* (2005).

Op grond van die verskillende algoritmes en verskeidenheid parameterinstellings, is dit 'n langdurige proses om handmatig te bepaal watter algoritme en instellings die beste resultate vir 'n gegewe taak lewer. Daarom is tydens die ontwikkeling van LIA outomaties met behulp van *Psearch* (Groenewald, 2006), 'n aanpassing van *Paramsearch* (Van den Bosch, 2005), bepaal watter algoritme die beste resultate lewer en ook wat die optimale parameterinstellings vir die betrokke datastel is. Daar is gevind dat die IB1-algoritme (Aha *et al.*, 1991) die beste resultate lewer. Die algoritme gebruik eienskapsvektore (*feature vectors*) om klasse aan woorde toe te ken en op grond van hierdie klasse kan bepaal word wat die lemma van die woord is (vgl. Groenewald, 2006 vir 'n beskrywing van die algoritme). Die eienskapsvektore en moontlike klasse moet versigtig gekies word om optimale resultate te verseker.

LIA is afgerig met 72 226 afrigtingsgevalle waarvan die helfte negatiewe afrigtingsdata is (d.i. woorde wat reeds lemmas is). Die negatiewe data is tot die afrigtingsdata gevoeg sodat LIA nie net leer hoe om woorde te lemmatiseer nie, maar ook wanneer om woorde te lemmatiseer. Die beste klassifiseerder is ontwikkel deur IB1 af te rig met data wat volgens eienskappe belyn is met veranderdewaardeverskilmetriek (*modified value difference metric*) as afstandsberekening, inligtingswinsgewigstoekenning (*information gain weighting*), as eienskapsgewig, 'n frekwensiedrempel (*frequency threshold*) van twee en inverse lineêre klasgewigte (*inverse linear class weights*). Met hierdie instellings behaal LIA 'n akkuraatheid van 92,8%.

LIA lewer goeie resultate en dus is besluit om ook van TiMBL gebruik te maak in die ontwikkeling van die fleksievormgenereerder

vir Afrikaans. Aangesien die kerntegnologie wat hier ontwikkel word presies die teenoorgestelde van LIA moet doen, is die fleksievormgenereerder AIL gedoop. Aangesien LIA en AIL so naverwant aan mekaar is, is besluit om LIA se afrigtingsdata aan te pas en te gebruik in die ontwikkeling van AIL. Daar word vervolgens dus kortliks uiteengesit hoe die eienskapsvektore en klasse van LIA gekies is, om te kan beskryf hoe LIA se afrigtingsdata lyk.

3.1 LIA se afrigtingsdata

Na verskeie eksperimente, is gevind dat die beste resultate vir outomatiese lemma-identifisering verkry word met vektore wat uit twintig eienskappe bestaan en wat volgens eienskappe belyn is (Groenewald, 2006). In Figuur 1 word voorbeelde gegee van eienskapsvektore wat uit 38 eienskappe bestaan, wat deur kommas geskei word, om die belyning daarvan te illustreer.

Figuur 1: Voorbeelde van eienskapsvektore uit LIA se afrigtingsdata

```
_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,v,e,r,n,i,e,t,i,g,b,a,r,e,Rre>ar

g,e,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,l,d,p,a,r,i,t,e,i,t,0

_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,k,o,e,r,s,v,e,r,s,k,i,l,l,e,Rle>

a,a,n,_,_,_,_,_,_,g,e,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,m,a,a,k,t,e,Mge>Rte>

o,p,_,_,_,_,_,_,_,g,e,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,s,t,a,a,n,Mge>

_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,o,p,s,l,a,e,0

g,e,_,_,_,_,_,e,s,t,e,s,g,e,s,o,n,d,h,e,i,d,s,h,e,r,s,i,e,n,i,n,g,s,r,a,d,e,Rde>ad
```

Tydens die ontwikkeling van LIA is verskeie eksperimente gedoen met die belyning van die eienskapsvektore. Daar is gevind dat regsbelynde data beter resultate lewer as linksbelynde data, maar dat die posisie van die verledetydsprefiks *ge-* soms foutiewe klassifikasie veroorsaak. Daarom is die data uiteindelik belyn op grond van eienskappe. As die letters *g-* en *-e-* aan die begin van ’n woord voorkom, vorm dit altyd die eerste twee eienskappe, aangesien dit moontlik die verledetydsvormende fleksieprefiks *ge-* kan wees. As dieselfde letters êrens binne ’n woord voorkom, is die *-g-* altyd die tiende eienskap en die *-e-* altyd die elfde eienskap, aangesien die fleksieaffiks *-ge-* tussen die twee konstituente van ’n skeibare werkwoord kan staan. Hierdie belyning geld net vir die karakterstring *-ge-,* aangesien dit die enigste Afrikaanse fleksie-

prefiks is en ook die enigste fleksieaffiks is wat nie noodwendig aan die linker- of regterkant van 'n woord hoef te staan nie.

Die laaste deel van die eienskapsvektor is die klas wat aan die betrokke woord toegeken is. Die klas bestaan uit drie dele, te wete die posisie van die affiks wat verwyder moet word, die letters wat verwyder moet word en die letters waarmee dit vervang moet word. Die posisie van die affiks word aangedui met 'n *L* wanneer die affiks 'n prefiks is wat aan die linkerkant van 'n woord voorkom, 'n *R* wanneer dit 'n suffiks is wat aan die regterkant van 'n woord voorkom en 'n *M* vir die spesiale geval van -*ge*- wat in die middel van 'n woord kan voorkom. Na die hoofletter wat aandui wat die posisie van die affiks is, volg die letters waaruit die affiks bestaan. In die voorbeelde in Figuur 1 is dit byvoorbeeld *re, le, ge, te* en *de.* Daarna volg 'n skerphakie (>) en die letters waarmee die affiks vervang moet word. In die laaste eienskapsvektor van Figuur 1 moet -*de* byvoorbeeld met -*ad* vervang word.

In die 72 226 woorde waarmee LIA afgerig is, is 268 verskillende klasse aan die woorde toegeken. Die omvang van die klasse is hoofsaaklik die gevolg van morfonologiese veranderings wat in die klasse hanteer word. Die meervoudsvorm *plase* sal byvoorbeeld nie net die klas *Re>* kry om aan te dui dat die meervouds -*e* verwyder moet word nie, maar eerder *Rse>as* sodat *plase* deur LIA verander kan word na *plaas.*

Aangesien daar reeds linguistiese navorsing vir die ontwikkeling van LIA se klasse gedoen is, is besluit om dieselfde klasse vir AIL te gebruik. As gevolg van die feit dat AIL egter die teenoorgestelde van LIA moet doen, is al die klasse ook omgedraai. Die lemma *plaas* sal byvoorbeeld vir die doeleindes van AIL met die klas *Ras>se* geannoteer word sodat die meervoudsvorm *plase* daaruit gemaak kan word. Die ontwikkeling van die Afrikaanse fleksievormgenereerder sal in die volgende afdeling in meer detail beskryf word.

4. Die ontwikkeling van 'n Afrikaanse fleksievormgenereerder

In 'n eerste poging om 'n fleksievormgenereerder vir Afrikaans te ontwikkel (AIL-1), is al die lemmas wat LIA as afvoer gee, as toevoervektore (*input vectors*) gebruik. Die omgekeerde van die klasse wat LIA gebruik het om 'n woord te lemmatiseer, is geneem as die klas van die betrokke lemma. LIA sou byvoorbeeld die klas *Rse>as* gebruik om *plaas* as afvoer te gee. Vir AIL is die afvoer (*plaas*) in 'n eienskapsvektor omgeskakel (wat lyk soos dié in Figuur 1) en die

klas *Ras>se* is daaraan toegeken. Dit is gedoen met elke woord in LIA se afrigtingsdata.

Aangesien dieselfde lemma egter meer as een fleksievorm kan hê (bv. *plase, plasie, plasies, geplaas, geplaasde*) het dit gebeur dat dieselfde eienskapsvektor in die AIL-data met verskillende klasse geannoteer is. Dit is verwarrend vir die masjienleeralgoritme en daarom is die verskillende klasse van dieselfde eienskapsvektore eers saamgevoeg om gekombineerde klasse te vorm. Die woord *plaas* sou dan geannoteer word met die gekombineerde klas *Ras>se;Ras>sie;Ras>sies;L>ge;L>geR>de.* Dieselfde klas sou ook aan die woord *haas* toegeken kon word, maar die woord *wolplaas* sal geannoteer word met ’n ander gekombineerde klas, naamlik *Ras>se;Ras>sie;Ras>sies,* aangesien *wolplaas* nie die verlede-tydsaffiks (*ge-*) of die deelwoordaffikse (*ge-...-de*) kan neem nie. Al die woorde wat in die LIA-data met klas 0 geannoteer is (negatiewe afrigtingsdata), is uitgehaal voordat die klasse gekombineer is, aangesien AIL nie die lemma van die woord as afvoer moet gee nie. Nadat die klasse gekombineer is, is ’n IB1-algoritme afgerig en geëvalueer. Dieselfde parameterinstellings wat die beste resultate gelewer het in die ontwikkeling van LIA is gebruik. Alhoewel die instellings van die lemma-identifiseerder nie noodwendig die beste resultate vir fleksievormgenerering sal lewer nie, is besluit om voorlopige eksperimente met hierdie instellings te doen. Parameter-optimalisering kan dan in die toekoms met behulp van *Psearch* gedoen word. Die resultate van die evaluasie word in die volgende afdeling bespreek.

4.1 Eerste poging: evaluasie

AIL-1 is op die afrigtingsdata geëvalueer met die *laat-een-uit-metode* (*leave one out*; vgl. Minnen *et al.*, 2000) en het in 45% van die gevalle die korrekte klas aan die betrokke lemma toegeken. Benewens die baie lae akkuraatheidsyfer, was baie van die klasse wat AIL-1 toegeken het, onvolledig. Hierdie onvolledige klasse resulteer uit die aard van LIA se afrigtingsdata, aangesien dit nie in LIA se geval belangrik was om volledige paradigmas by die afrigtingsdata in te sluit nie. Die meervoudsvorm van die woord *das* is byvoorbeeld by LIA se afrigtingsdata ingesluit, maar die diminutiefen die meervoud van die diminutiefvorme van *das* is nie deel van LIA se afrigtingsdata nie. Nou word *das* deur AIL-1 geannoteer met die klas *R>se,* maar dit moet eintlik geannoteer word met die klas *R>se;R>sie;R>sies.* AIL-1 is dus nie in staat daartoe om alle moontlike fleksievorme vir ’n lemma te genereer nie.

Verder lei die onvolledige klasse ook tot baie ekstra klasse wat die gevalruimte (moontlike annotasies) vergroot en tot laer akkuraatheid lei. Die woord *das* kom byvoorbeeld in AIL-1 se afrigtingsdata voor met die klas *R>se*, terwyl *jas* die volledige gekombineerde klas *R>se;R>sie;R>sies* het, omdat *jasse, jassie* en *jassies* deel was van LIA se afrigtingsdata. Alhoewel dié twee woorde dus eintlik in dieselfde klas hoort, veroorsaak die aard van LIA se afrigtingsdata 'n vergroting in die gevalruimte, omdat die twee woorde (volgens die afrigtingsdata) nie met dieselfde klas geannoteer behoort te word nie.

Aangesien AIL-1 nie bevredigende resultate gelewer het nie, is besluit om 'n ander strategie te volg om 'n volgende fleksievormgenereerder te ontwikkel. Die ontwikkeling van AIL-2 word in die volgende afdeling beskryf.

4.2 Tweede poging: volledige paradigmas

Aangesien die grootste probleem met AIL-1 die onvolledige klasse is wat tot laer akkuraatheid gelei het, is besluit om die paradigmas in AIL-2 se afrigtingsdata te voltooi deur seker te maak dat die gekombineerde klasse wat aan woorde toegeken is, volledig is. Daar is besluit om 5 000 woorde se klasse na te gaan en te voltooi. Hierdie 5 000 woorde kon dan gebruik word om 'n klassifiseerder af te rig waarna die afrigtingsdata met behulp van skoenlussteekproefneming (*bootstrapping*; Abney, 2002) uitgebrei kon word totdat bevredigende resultate met die fleksievormgenereerder verkry is.

Dit het 'n Afrikaanse linguis 'n week geneem om die klasse van die 5 000 woorde na te gaan en te voltooi waar nodig. Uiteindelik was daar 454 verskillende klasse vir die 5 000 woorde. Dit beteken dat die gevalruimte baie groot is en dat die akkuraatheid van die uiteindelike klassifiseerder noodwendig laer sal wees as dié van LIA met slegs 268 klasse.

AIL-2 is op dieselfde manier geëvalueer as AIL-1 en op die afrigtingsdata het hierdie klassifiseerder 'n akkuraatheid van 60% gehad. Aangesien die klassifiseerder met net 5 000 woorde afgerig is, was 'n lae akkuraatheidsyfer te wagte en daarom is 1 000 nuwe gevalle daarmee geklassifiseer as eerste herhaling in die skoenlussteekproefnemingsproses. Hierdie data is met die hand deurgegaan met die doel om die verkeerde klassifikasies wat daarin voorkom, reg te maak sodat die 1 000 nuwe woorde by die 5 000 woorde van die afrigtingsdata gevoeg kan word. 'n Nuwe klassifiseerder sou met die resulterende 6 000 woorde afgerig word en 1 000 nuwe woorde sou

dan met die nuwe klassifiseerder afgerig word. Hierdie proses sou herhaal word totdat die klassifiseerder bevredigende resultate lewer.

Op die 1 000 nuwe woorde was AIL-2 egter net 29,4% akkuraat. Hierdie lae akkuraatheidsyfer was grotendeels te wyte aan die feit dat daar 93 klasse in die nuwe data voorgekom het wat nie in die afrigtingsdata voorgekom het nie. Dit was dus op hierdie stadium duidelik dat ’n fleksievormgenereerder wat op hierdie manier ontwikkel word, baie afrigtingsdata sou nodig hê voordat dit bevredigende resultate kon lewer. Aangesien die generering van afrigtingsdata ’n duur proses is, is besluit om eerder ’n ander metode te gebruik in die ontwikkeling van ’n fleksievormgenereerder vir Afrikaans. Die metode wat gebruik is om AIL-3 te ontwikkel, word in die afdeling hieronder beskryf.

4.3 Aparte klassifiseerders

Aangesien die akkuraatheid van enige masjienleeralgoritme wat met relatief min data afgerig is, afhanklik is van die grootte van die gevalruimte (hoeveelheid moontlike klasse) van daardie klassifiseerder, is besluit om die gevalruimte van die fleksievormgenereerder so klein as moontlik te maak. Daar is tydens die ontwikkeling van AIL-1 en AIL-2 eintlik van een klassifiseerder verwag om twaalf verskillende take te verrig – een vir elke soort fleksievorm wat moontlik gegenereer kan word. Deur twaalf verskillende klassifiseerders af te rig, word die gevalruimte vir elke klassifiseerder drasties verklein en daarom behoort hierdie aparte klassifiseerders beter te vaar as een klassifiseerder.

Die data wat gebruik is om AIL-1 af te rig, is daarom volgens affiks in twaalf dele verdeel en gebruik om twaalf verskillende klassifiseerders af te rig. Die minimalisering van die gevalruimte blyk duidelik in die tweede kolom in Tabel 4 met meervoud (PL) wat met 106 moontlike klasse die grootste gevalruimte van al die klassifiseerders het. Dit is ’n merkbare verskil wanneer die 454 moontlike klasse van AIL-2 in gedagte gehou word. Die kategoriename staan, in volgorde, vir die volgende fleksieaffikse en kombinasies van fleksieaffikse (wat in een stap verwyder kan word): attributiewe *-e*, komparatief, diminutief, meervoud en diminutief, deelsgenitiewe *-s*, partisipium (deelwoord), partisipium en attributiewe *-e,* partisipium en komparatief, partisipium en superlatief, meervoud, verlede tyd en superlatief.

Die aparte klassifiseerders is weereens met die laat-een-uit-metode op die afrigtingsdata geëvalueer en die resultate van hierdie evaluasie word in die vierde kolom van Tabel 4 weergegee.

Tabel 4: AIL-3: aparte klassifiseerders

Klassifiseerder	Aantal klasse in afrigtings-data	Afrigtings gevalle in afrigtings-data	Akkuraatheid (afrigtingsdata)	Akkuraatheid (nuwe data)
ATT	65	5 813	97,33%	94,30%
COMP	38	936	90,38%	94,30%
DIM	41	1 343	89,20%	92,02%
DIM_PL	38	720	84,42%	88,77%
GEN	1	36	**100,00%**	**78,61%**
PRTC	11	133	**93,23%**	**54,46%**
PRTC_ATT	15	1 329	**69,45%**	**81,19%**
PRTC_COMP	1	2	**100,00%**	**50,49%**
PRTC_SUP	5	6	**33,33%**	**20,79%**
PL	106	20 391	89,25%	95,12%
PST	6	2 947	90,43%	86,13%
SUP	3	977	99,39%	99,36%

Om die resultate te verifieer wat op die afrigtingsdata verkry is, is AIL-3 ook geëvalueer op ’n klein hoeveelheid lukraakgeselekteerde data wat nie deel was van die afrigtingsdata nie. Die nuwe toetsdata is met woordsoortetikette geannoteer en konjunkte, tussenwerpsels, voorsetsels en ander woorde uit geslote woordsoortkategorieë is uit die lys verwyder. Die resulterende data (1 000 woorde) is op grond van woordsoortkategorieë in drie datastelle verdeel, bestaande uit 158 adjektiewe, 103 werkwoorde en 739 naamwoorde elk. Die verskillende groottes van hierdie drie datastelle is die gevolg van die feit dat die toetsdata lukraak geselekteer is. Woorde met meerduidigheid wat woordsoort betref, is geannoteer met die mees prototipiese kategorie. Die naamwoorddatastel is gebruik om die PL-, DIM-, en DIM_PL-klassifiseerders te evalueer; die adjektiewe vir die

evaluasie van die ATT-, COMP-, GEN-, en SUP-klassifiseerders en die werkwoorddatastel is gebruik om die PRTC-, PRTC_ATT-, PRTC_COMP-, PRTC_SUP- en PST-klassifiseerders te evalueer.

Die resultate van hierdie evaluasie word in kolom 5 van Tabel 4 gegee. Met die uitsondering van vier klassifiseerders (dié resultate is in vetdruk en sal in afdeling 5 bespreek word), lewer die klassifiseerders van AIL-3 ook bevredigende resultate op nuwe data. Om die gemiddelde akkuraatheid van AIL-3 te bereken, is die toetsdatastelle van die verskillende klassifiseerders bymekaar gesit en die persentasie korrekgeklassifiseerde woorde van dié datastel is bereken. AIL-3 behaal op hierdie nuwe datastel ’n akkuraatheid van 86,88%.

Alhoewel die resultate belowend lyk, is AIL-3 waarskynlik nog nie akkuraat genoeg om in ’n MTT-sisteem geïmplementeer te word nie. AIL-3 voldoen ook tans net aan een van die vereistes wat vir ’n fleksievormgenereerder gestel is, aangesien dit ’n spesifieke fleksievorm van ’n gegewe lemma kan genereer deur die lemma na net een van die klassifiseerders te stuur. In die volgende afdeling word eers maniere bespreek om die akkuraatheid van die AIL-3-klassifiseerders te verhoog om die fleksievormgenereerder sodoende meer implementeerbaar te maak. Daarna word die moontlikheid van ’n voorverwerkingsmodule wat lemmas outomaties na die aangewese klassifiseerders stuur kortliks bespreek, sodat alle fleksievorme daarvan gegenereer kan word.

5. AIL-3: verbeteringsmoontlikhede

Al die klassifiseerders van AIL-3 is vir die doeleindes van die eksperimente wat hier bespreek word, afgerig met die parameterinstellings wat die beste resultate gelewer het in die ontwikkeling van LIA. Parameteroptimalisering moet daarom nog vir elkeen van die aparte klassifiseerders gedoen word om te verseker dat die beste moontlike resultate verkry word met die afrigtingsdata wat beskikbaar is. Sodoende kan die klassifiseerders “outomaties” verbeter word sonder dat verdere afrigtingsdata handmatig ontwikkel moet word.

Die aard van en gebrek aan afrigtingsdata lei egter nog in die geval van sommige van die klassifiseerders tot ’n lae akkuraatheidsyfer en addisionele afrigtingsdata moet vir hierdie klassifiseerders ontwikkel word. Die PRTC_SUP-klassifiseerder vaar byvoorbeeld nie goed op die afrigtingsdata of op die nuwe data nie. Dit is waarskynlik te wyte aan die feit dat die klassifiseerder met slegs ses gevalle afgerig is en vyf moontlike klasse het. Aangesien daar dus gemiddeld net

meer as een afrigtingsgeval vir elkeen van hierdie klasse is, is dit vir die klassifiseerder bykans onmoontlik om te bepaal wat die klas van ’n ongesiene geval moet wees. Dié klassifiseerder kan daarom relatief maklik verbeter word deur addisionele afrigtingsdata tot die reeds bestaande ses gevalle toe te voeg.

Die ander klassifiseerders waarmee partisipiumvorms gegenereer moet word (PRTC, PRTC_ATT, PRTC_COMP en PRTC_SUP), behaal nie hoë akkuraatheidsyfers op die nuwe data nie – die hoogste is PRTC_ATT wat slegs 81,19% akkuraat is. Dit is in die eerste plek te wyte aan die klein hoeveelhede afrigtingsdata wat vir hierdie klassifiseerders beskikbaar is. By nadere ondersoek blyk dit ook dat die afrigtingsdata min gevalle bevat waar die affiks *-ge-* in die middel van ’n woord ingevoeg moet word (soos by *aan**ge**haalde, op**ge**wekter,* ensovoorts), terwyl daar in die nuwe data verskeie lemmas is wat sodanig geïnflekteer moet word. Deur sulke voorbeelde tot die afrigtingsdata toe te voeg, kan die akkuraatheid van hierdie klassifiseerders ook verbeter word.

Die verskil tussen die akkuraatheid van die GEN-klassifiseerder op die afrigtingsdata en die nuwe data, is ook te wyte aan die aard van die afrigtingsdata. Aangesien LIA se negatiewe afrigtingsdata weggelaat is tydens die voorbereiding van die afrigtingsdata vir die aparte klassifiseerders, is slegs adjektiewe wat ’n deelsgenitiewe *-s* moet kry by die afrigtingsdata van die GEN-klassifiseerder ingesluit. Wanneer die afgerigte GEN-klassifiseerder gekonfronteer word met adjektiewe wat nié ’n deelsgenitiewe *-s* kan neem nie, word die enigste klas wat aan die klassifiseerder bekend is (R>s wat ’n -s toevoeg aan die einde van ’n woord) aan dié woord toegeken. Die klassifiseerder verander dus byvoorbeeld *skaars* in **skaarss.* Hierdie probleem kan opgelos word deur negatiewe afrigtingsdata wat met ’n klas *0* gemerk is (soos *skaars*), tot die bestaande afrigtingsdata toe te voeg.

Dit is egter waarskynlik nie net hierdie klassifiseerder wat negatiewe afrigtingsdata moet bykry nie, aangesien die ander klassifiseerders ook net met positiewe afrigtingsdata afgerig is. PL-klasse mag egter net toegeken word aan naamwoorde en die adjektief *mooi* moet dus eintlik deur die PL-klassifiseerder met klas *0* gemerk word. Tans kan die PL-klassifiseerder nie ’n klas *0* toeken nie en dit sal een of ander PL-klas aan *mooi* toeken wanneer dit daarmee gekonfronteer word. Dieselfde geld ook vir al die ander klassifiseerders.

Negatiewe afrigtingsdata vergroot egter die gevalruimte van ’n betrokke klassifiseerder en dit sou dus wenslik wees om die probleem

op ’n ander manier op te los. Aangesien spesifieke fleksieaffikse per definisie slegs met spesifieke woordsoorte verbind, kan woordsoortinligting gebruik word om te bepaal watter woorde na ’n spesifieke klassifiseerder gestuur moet word. Aangesien ’n Afrikaanse woordsoortetiketteerder reeds by CTexT ontwikkel is, is dit moontlik om woordsoortetikette aan woorde toe te ken. Daar kan dus ’n reëlgebaseerde voorverwerkingsmodule tot AIL-3 toegevoeg word wat op grond van die woordsoort van ’n woord bepaal watter klassifiseerders ’n woord moet klassifiseer. Die voorverwerkingsmodule sal byvoorbeeld ’n naamwoord na die PL-, DIM- en DIM_PL-klassifiseerders stuur, ’n adjektief na die ATT-, GEN-, COMP- en SUP-klassifiseerders en ’n werkwoord na die PRTC-, PRTC_ATT-, PRTC_COMP-, PRTC_SUP- en PST-klassifiseerders.

So ’n voorverwerkingsmodule bring ook mee dat AIL-3 aan die twee gestelde vereistes vir ’n Afrikaanse fleksievormgenereerder voldoen. Dit is moontlik om (vir doeleindes van die MV-sisteem) ’n spesifieke fleksievorm van ’n woord te genereer, aangesien die woord net na die aangewese klassifiseerder gestuur kan word. Met die voorverwerkingsmodule sou AIL-3 egter ook al die fleksievorme van ’n betrokke woord kan genereer vir die doeleindes van ALEXANDER.

Dit is vervolgens ook moontlik om slegs enkele klassifiseerders in toepassings te implementeer. CTexT is tans besig met die ontwikkeling van ’n grammatikatoetser vir Afrikaans. Een van die grammatikale konstruksies wat deur die grammatikatoetser hanteer moet word, is naamwoord-telwoord-kongruensie. Die grammatikatoetser moet egter nie net kan bepaal dat ’n konstruksie soos *Die vier donkie* ’n grammatikale fout is nie, maar moet ook ’n goeie verbetering kan voorstel vir die fout wat die gebruiker gemaak het. Die PL-klassifiseerder kan hier gebruik word om die juisste meervoudsvorm van die naamwoord voor te stel.

6. Samevatting

In hierdie artikel is die ontwikkeling van ’n fleksievormgenereerder (AIL) vir Afrikaans beskryf. Aangesien AIL ontwikkel is met twee verskillende projekte in gedagte, is twee vereistes gestel waaraan dit moet voldoen: AIL moet in staat wees om slegs een spesifieke fleksievorm van ’n lemma te genereer, maar ook om alle moontlike fleksievorme van ’n lemma te genereer. Daar is besluit om die fleksievormgenereerder te ontwikkel met behulp van masjienleertegnieke, aangesien ’n vorige projek reeds bewys het dat die reëlgebaseerde benadering nie goeie resultate lewer vir Afrikaanse morfo-

logiese analise nie. Die algoritme wat in die ontwikkeling gebruik is, is IB1 wat deel is van TiMBL.

Daar is met drie verskillende metodes geëksperimenteer in die ontwikkeling van AIL, maar eindelik het dit geblyk dat aparte klassifiseerders die beste resultate lewer. Dit is ook makliker om aparte klassifiseerders te verbeter deur spesifieke afrigtingsdata tot die klassifiseerders toe te voeg. Met behulp van 'n reëlgebaseerde voorverwerkingsmodule voldoen AIL-3 aan die twee vereistes wat gestel is. Die gemiddelde akkuraatheid van AIL-3 (wat bestaan uit 12 aparte klassifiseerders) op die afrigtingsdata is 86,37% en in 'n evaluasie met 'n klein hoeveelheid nuwe data behaal AIL-3 'n gemiddelde akkuraatheid van 86,88%. Alhoewel die resultate nog nie na wense is nie, kan die aparte klassifiseerders relatief maklik verbeter word om die gemiddelde akkuraatheid van AIL-3 te verhoog.

AIL-3 sal daarom verbeter word deur die optimale parameterinstellings van elke klassifiseerder te vind. Addisionele afrigtingsdata sal ook vir sommige klassifiseerders geannoteer word om sodoende die akkuraatheid daarvan te verhoog. Dit sou ook interessant kon wees om met ander masjienleertegnieke soos besluitnemingsbome (*decision trees*) te eksperimenteer om te bepaal of IB1 inderdaad die beste masjienleertegniek is om te gebruik.

7. Erkenning

Handré Groenewald, Martin Puttkammer en Gerhard van Huyssteen het waardevolle bydraes gelewer tydens die konseptualisering van AIL. Dankie ook aan Handré en Martin vir hulp met die uitvoer van die eksperimente wat hier beskryf word.

Geraadpleegde bronne

ABNEY, S. 2002. Bootstrapping. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 6-12 July 2002, Philadelphia. p. 360-367.

AHA, D., KIBLER, D. & ALBERT, M. 1991. Instance-based learning algorithms. *Machine learning*, 6:37-66.

ARMSTRONG, S. 1996. Multext: multilingual text tools and corpora. (*In* Feldweg, H. & Hinrichs, E.W., *Reds*. Lexikon und Text. Tübingen: Max Niemeyer. S. 107-112.)

BEESLEY, K.R. 1989. Computer analysis of Arabic morphology: a two-level approach with detours. (*In* Comrie, B. & Eid, M., *eds*. Perspectives on Arabic Linguistics III: Papers from the 3rd Annual Symposium on Arabic Linguistics. Amsterdam: Benjamins. p. 155-172.)

BEESLEY, K.R. 1990. Finite-state description of Arabic morphology. Proceedings of the 2nd Cambridge Conference on Bilingual Computing in Arabic and English, Cambridge.

BLACK, A., RITCHIE, G., PULMAN, S. & RUSSELL, G. 1987. Formalisms for morphographemic description. Proceedings of the 3rd Conference of the European Chapter of the Association for Computational Linguistics, Copenhagen. p. 11-18.

CARTER, D. 1995. Rapid development of morphological descriptions for full language processing systems. Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, Dublin. p. 202-209.

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

DAELEMANS, W., VAN DEN BOSCH, A., ZAVREL, J. & VAN DER SLOOT, K. 2004. TiMBL: Tilburg memory based learner, version 5.1: reference guide. (ILK technical report 04-02.) http://ilk.uvt.nl/downloads/pub/papers/ilk0402.pdf Date of access: 26 Sept. 2007.

DIRIX, P., VANDEGHINSTE, V. & SCHUURMAN, I. 2005. METIS: example-based machine translation using monolingual corpora – system description. Proceedings of the Workshop on Example-based Machine Translation. Available: MT SUMMIT X. Phuket, Thailand. http://www.ccl.kuleuven.be/Papers/Dirix.pdf Date of access: 12 Sept. 2007.

GROENEWALD, H.J. 2006. Automatic lemmatisation for Afrikaans. Potchefstroom: Noordwes-Universiteit. (M.Ing.-verhandeling.)

KARTTUNEN, L. 1983. Kimmo: a general morphological processor. (*In* Dalrymple, M., Doron, E., Goggin, J., Goodman, B. & McCarthy, J., *eds*. *Texas linguistic forum*, 22:165-[186].)

KOSKENNIEMI, K. 1983. Two-level morphology: a general computational model for word-form recognition and production. Helsinki: University of Helsinki, Department of General Linguistics. (Publication 11.)

KOSKENNIEMI, K. 1986. Compilation of automata from morphological two-level rules. (*In* Karlsson, F., *ed*. Papers from the 5th Scandinavian Conference on Computational Linguistics, Helsinki. p. 143-149.)

MINNEN, G., BOND, F. & COPESTAKE, A. 2000. Memory-based learning for article generation. Proceedings of the 4th Conference on Computational Natural Language Learning and of the 2nd Learning Language in Logic workshop, Lisbon. p. 43-48.

RITCHIE, G., BLACK, A., PULMAN, S. & RUSSELL, G. 1987. The Edinburgh/Cambridge morphological analyser and dictionary system (version 3.0) user manual. Edinburgh: University of Edinburgh. (Technical report software paper, no. 10.)

RITCHIE, G., RUSSELL, G., BLACK, A. & PULMAN, S. 1992. Computational morphology: practical mechanisms for the English lexicon. Cambridge: MIT.

VAN DEN BOSCH, A. 2005. Paramsearch 1.0 beta patch 24. http://ilk.uvt.nl/software.html#paramsearch Date of access: 20 Jun. 2007.

Kernbegrippe:

Afrikaanse taalkunde
fleksievormgenereerder
kerntegnologieë
lemma-identifiseerder
masjienleer

Key concepts:

Afrikaanslinguistics
core technologies
inflected form generator
lemmatiser
machine learning