Enkele tegnieke vir die ontwikkeling en benutting van etiketteringhulpbronne vir hulpbronskaars tale

AC Griebenow

12605883

Verhandeling voorgelê ter gedeeltelike nakoming vir die graad *Magister Scientiae* in Rekenaarwetenskap aan die Potchefstroomkampus van die Noordwes-Universiteit

Studieleier: Prof GR Drevin

Medestudieleier: Mnr DP Snyman

Oktober 2015

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

ERKENNINGS

Ek wil graag opreg dankie sê aan

- My Vader in die hemel. Dankie vir die geleenthede en die verstandelike vermoë wat ek onverdiend ontvang het. Ek dank U vir alles wat ek kon leer en vir die krag om hierdie studie te kon voltooi.
- My ouers, Hans en Jeanine van Dijkhorst. Dankie dat Pa en Ma my van kleins af ondersteun en in my glo. Dankie vir die harde werk en opoffering, sodat ek kon studeer. Stappie vir stappie het ek uiteindelik daar uitgekom.
- My man, Gideon Griebenow. Dankie vir jou deurlopende ondersteuning en hulp. Sonder jou sou ek dit nie kon doen nie.
- Familie en vriende. Dankie vir al die ondersteuning en belangstelling.
- Suzanne Erasmus en Bea Basson, julle is vriendinne so reg uit die hemel. Suzanne, dankie vir al jou hulp met die kinders, sodat ek kon werk. Bea, al is jy vêr, is jou hart altyd naby.
- Prof. Drevin. Dankie vir die geleentheid en bereidwilligheid om my studieleier te wees.
- Dirk Snyman. Dankie vir waardevolle insette in die samestelling van hierdie verhandeling.

OPSOMMING

ENKELE TEGNIEKE VIR DIE ONTWIKKELING EN BENUTTING VAN ETIKETTERINGHULPBRONNE VIR HULPBRONSKAARS TALE

AC GRIEBENOW

Die ontwikkeling van hulpbronne in enige taal is ’n duur proses en juis om hierdie rede bestaan daar hulpbronskaars tale, waaronder ook die inheemse tale van Suid-Afrika, wat nie oor voldoende etiketteringhulpbronne beskik nie. Tegnieke en metodes vir die optimering van beskikbare hulpbronne en die verbetering in die akkuraatheid van ’n woordsoortetiketteerder word in hierdie studie ondersoek en ontgin met Afrikaans wat dien as hulpbronskaars taal. Die studie het ten doel om

i) te bepaal of kombinasietegnieke effektief gebruik kan word om die akkuraatheid van ’n Afrikaanse woordsoortetiketteerder te verbeter, en
ii) te bepaal of strukturele gedeeltelik-gekontroleerde leer effektief aangewend kan word om die akkuraatheid van ’n gekontroleerde leer Afrikaanse woordsoortetiketteerder te verbeter.

Om die eerste doelwit te bereik word ’n ondersoek geloods na bestaande metodes vir die kombinering van klassifikasie-algoritmes waarna vier woordsoortetiketteerders, wat onderskeidelik deur middel van MBT, SVM*light*, MXPOST en TnT afgerig is, in ’n kombinasie-woordsoortetiketteerder saamgevoeg word deur gebruik te maak van geweegde stemming. Gewigte word bereken deur gebruik te maak van totale presisie, etiketpresisie en ’n kombinasie van presisie en herroeping. Die kombinasie van woordsoortetiketteerders toon nie vir alle gevalle ’n foutkoersreduksie met betrekking tot die basislyn nie, maar slaag wel daarin om vir sommige gevalle ’n foutkoersreduksie van tot 18.48% te behaal.

Om die tweede doelwit te bereik word ’n ondersoek geloods na bestaande gedeeltelik-gekontroleerde leeralgoritmes met spesifieke klem op strukturele gedeeltelik-gekontroleerde leer. Strukturele gedeeltelik-gekontroleerde leer word deur middel van die SVD-ASO-algoritme geïmplementeer waartydens die gedeelde struktuur met behulp van hulpprobleme uit ongeëtiketteerde data onttrek word alvorens ’n woordsoortetiketteerder afgerig word. Die aanwending van ongeëtiketteerde data tydens die afrigting van ’n woordsoortetiketteerder het ’n foutkoersreduksie tot die basislyn van tot 1.67%.

Alhoewel die foutkoersreduksie nie statisties betekenisvol blyk te wees vir alle gevalle nie, toon die resultate dat dit wel moontlik is om die akkuraatheid vir sommige gevalle te verhoog.

Sleutelterme:

hulpbronskaars taal, masjienleer, gedeeltelik-gekontroleerde leer, strukturele leer, mensetaaltegnologie, kombinasie-woordsoortetiketteerder, natuurliketaalverwerking.

ABSTRACT

DIFFERENT TECHNIQUES FOR THE DEVELOPMENT AND UTILISATION OF TAGGING RESOURCES FOR RESOURCE-SCARCE LANGUAGES

AC GRIEBENOW

Because the development of resources in any language is an expensive process, many languages, including the indigenous languages of South Africa, can be classified as being resource scarce, or lacking in tagging resources. This study investigates and applies techniques and methodologies for optimising the use of available resources and improving the accuracy of a tagger using Afrikaans as resource-scarce language and aims to

i) determine whether combination techniques can be effectively applied to improve the accuracy of a tagger for Afrikaans, and

ii) determine whether structural semi-supervised learning can be effectively applied to improve the accuracy of a supervised learning tagger for Afrikaans.

In order to realise the first aim, existing methodologies for combining classification algorithms are investigated. Four taggers, trained using MBT, SVM*light*, MXPOST and TnT respectively, are then combined into a combination tagger using weighted voting. Weights are calculated by means of total precision, tag precision and a combination of precision and recall. Although the combination of taggers does not consistently lead to an error rate reduction with regard to the baseline, it manages to achieve an error rate reduction of up to 18.48% in some cases.

In order to realise the second aim, existing semi-supervised learning algorithms, with specific focus on structural semi-supervised learning, are investigated. Structural semi-supervised learning is implemented by means of the SVD-ASO-algorithm, which attempts to extract the shared structure of untagged data using auxiliary problems before training a tagger. The use of untagged data during the training of a tagger leads to an error rate reduction with regard to the baseline of 1.67%.

Even though the error rate reduction does not prove to be statistically significant in all cases, the results show that it is possible to improve the accuracy in some cases.

Key terms:

resource-scarce language, machine learning, semi-supervised learning, structural learning, human language technology, combination tagger, natural language processing.

INHOUDSOPGAWE

Hoofstuk 1 Inleiding ... 1
1.1 Agtergrond ... 1
1.2 Probleemstelling ... 5
1.3 Navorsingsvrae ... 6
1.4 Doelstellings ... 6
1.5 Sentrale Teoretiese Stelling ... 6
1.6 Navorsingsmetodes ... 7
1.6.1 Literatuurstudie ... 7
1.6.2 Eksperimentering ... 8
1.6.3 Presisie, herroeping en akkuraatheid ... 8
1.6.4 Foutkoers en foutkoersreduksie ... 9
1.6.5 McNemar se toets ... 10
1.7 Ontplooiing van navorsing ... 10
Hoofstuk 2 Inleiding tot leer- en kombinasiemetodes ... 12
2.1 Masjienleer ... 12
2.1.1 Gekontroleerde leer ... 13
2.1.1.1 Klassifikasie ... 13
2.1.2 Ongekontroleerde leer ... 19
2.1.3 Gedeeltelik-gekontroleerde leer ... 20
2.1.3.1 Selfleer ... 21
2.1.3.2 Deelleer ... 21
2.1.3.3 Generatiewe modelle ... 22
2.1.3.4 Gedeeltelik-gekontroleerde Steunvektorklassifiseerders ... 23
2.1.3.5 Grafiekgebaseerde metodes ... 26
2.2 Kombinasie ... 28
2.2.1 Kaskademetode ... 29
2.2.2 Stemmetode ... 29
2.2.2.1 Eenvoudige stemming ... 30
2.2.2.2 Geweegde stemming ... 30

2.2.3 Versterking ... 32
2.2.4 Groepering ... 32
2.2.5 Opstapeling ... 34
2.2.5.1 Opgestapelde waarskynlikheidstemming ... 35
2.2.5.2 Geweegde waarskynlikheidsverdelingstemming ... 36
2.2.6 Klassifiseerderomruiling ... 38
2.2.7 Volgorde sisteem ... 38
2.2.8 Interne kombinasie ... 39
2.3 Samevatting ... 40
Hoofstuk 3 ’n Afrikaanse Kombinasie-woordsoortetiketteerder ... 42
3.1 Metode ... 42
3.2 Eerstevlakklassifiseerders ... 42
3.2.1 MBT ... 42
3.2.2 SVM^{light} ... 43
3.2.3 MXPOST ... 44
3.2.4 TnT ... 45
3.3 Kombinasieklassifiseerder ... 46
3.4 Resultate ... 49
3.4.1 Kwaliteit van eerstevlakklassifiseerders ... 49
3.4.2 Ooreenkomste tussen eerstevlakklassifiseerders ... 50
3.4.3 Die teorie van kombinasie ... 51
3.4.4 Kombinasie in detail ... 52
3.4.5 Resultate van kombinasie ... 55
3.4.6 Statistiese betekenisvolheid ... 58
3.4.7 Die effek van ’n vereenvoudigde etiketstel ... 60
3.5 Samevatting ... 61
Hoofstuk 4 Strukturele gedeeltelik-gekontroleerde leer ... 63
4.1 Strukturele leer ... 63
4.2 ’n Model vir die leer van strukture ... 63
4.2.1 Standaard liniêre model ... 64
4.2.2 Liniêre model vir strukturele leer ... 65
4.2.3 Wisselstruktuuroptimeringsalgoritme ... 66

4.3 Eksperimentele Opstelling 67
4.3.1 Hulpprobleme 70
4.3.2 Die generering van die struktuurmatriks 74
4.3.2.1 Parameters 74
4.3.2.2 Uitvoer 74
4.3.2.3 Algoritme 74
4.3.3 Die afrigting van ’n klassifiseerder 76
4.3.4 Die klassifiseerder 79
4.3.5 Resultate 79
4.3.6 Statistiese betekenisvolheid 80
4.4 Samevatting 82
Hoofstuk 5 Slot 83
5.1 Samevatting 83
5.2 Gevolgtrekking 85
5.3 Vooruitskouing 88
5.4 Slot 88
Bronnelys 90
Bylaag A: Engelse terme 99
Bylaag B: Woordsoortetikette 103
Bylaag C: 1000 mees frekwente tekseenhede 106

LYS VAN FIGURE

Figuur 1.1 – Hulpbronne as ’n voorvereiste vir die ontwikkeling van toepassings 1
Figuur 2.1 – Die gekontroleerde leerprobleem 13
Figuur 2.2 – Steunvektorklassifiseerdersbeslissingsgrens. 24
Figuur 2.3 – S3VM beslissingsgrens. 24
Figuur 2.4 – ’n Grafiek wat gekonstrueer is uit geëtiketteerde en ongeëtiketteerde klassifikasiegevalle. . 27
Figuur 3.1 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 1 47
Figuur 3.2 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 2 47
Figuur 3.3 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 3 47
Figuur 3.4 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 4 48
Figuur 3.5 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 5 48
Figuur 4.1 – Onttrekking van hulprye 71
Figuur 4.2 – Pseudokode vir generering van die struktuurmatriks 76
Figuur 4.3 – Gewig- en eienskapvektore 77
Figuur 4.4 – Pseudokode vir die afrigting van ’n voorspeller 78
Figuur 4.5 – Pseudokode vir klassifiseerder 79
Figuur 5.1 – Die werklike akkuraatheid van die kombinasie-woordsoortetiketteerder in verhouding tot minimum en maksimum 86

LYS VAN TABELLE

Tabel 1.1 – Verwarringsmatriks vir werklike en voorspelde klassifikasies 9
Tabel 1.2 – Verwarringsmatriks vir klassifikasies van Klassifiseerder A en Klassifiseerder B 10
Tabel 2.1 – Komplekse eienskapvektore 16
Tabel 2.2 – Die naïewe Bayes-model op woordsoortetiket: ASA 17
Tabel 2.3 – Klassifikasiegeval 17
Tabel 2.4 - ’n Voorbeeld van ’n WPDV-klassifikasiegeval 37
Tabel 3.1 – Presisie-Herroeping stemmetode voorbeeld 48
Tabel 3.2 – Akkuraatheid van eerstevlakklassifiseerders op UVVM en TEST 50
Tabel 3.3 – Patrone van ooreenkomste tussen die vier klassifiseerders op UVVM en TEST 50
Tabel 3.4 – Beraamde akkuraathede van toenemende vlakke van sukses 51
Tabel 3.5 – Beraamde akkuraathede van toenemende vlakke van sukses - TnT uitgesluit 52
Tabel 3.6 – Tekseenhede in TEST wat dikwels verkeerd geëtiketteer is 53
Tabel 3.7 – Etiketpare wat dikwels met mekaar verwar word met die aantal foutvoorkomste 54
Tabel 3.8 – Presisie en herroeping vir die etikette wat dikwels verwarring veroorsaak 55
Tabel 3.9 – Resultate van die kombinasie van die eerstevlakklassifiseerders d.m.v. verskillende stemmetodes 55
Tabel 3.10 – Resultate van die kombinasie van Kombi 3 d.m.v. verskillende stemmetodes 56
Tabel 3.11 – Die kombinasie van Kombi 3 in vergelyking met TnT 57
Tabel 3.12 – Die verskil tussen TnT en kombinasieklassifiseerders op UVVM 58
Tabel 3.13 – Die verskil tussen TnT en kombinasieklassifiseerders op TEST 58
Tabel 3.14 – Die verskil tussen MBT en Kombi 3 op UVVM 59
Tabel 3.15 – Die verskil tussen MBT en Kombi 3 op TEST 59
Tabel 3.16 – Die verskil tussen TnT en Kombi 3 op UVVM 60
Tabel 3.17 – Die verskil tussen TnT en Kombi 3 op TEST 60
Tabel 3.18 – Akkuraatheid gemeet volgens die volledige etiketstel en slegs die hoofkategorieë 61
Tabel 4.1 – Voorstelling van eienskapvektore e1 tot e400000 69
Tabel 4.2 – Voorbeeld van die ongekontroleerde strategiehulpprobleme 72
Tabel 4.3 – Voorbeeld van die gedeeltelik-gekontroleerde strategiehulpprobleme 73
Tabel 4.4 – Woordsoortetikettering akkuraatheid en foutkoersreduksie 80
Tabel 4.5 – Die verskil tussen gekontroleerde en gedeeltelik-gekontroleerde leer 81

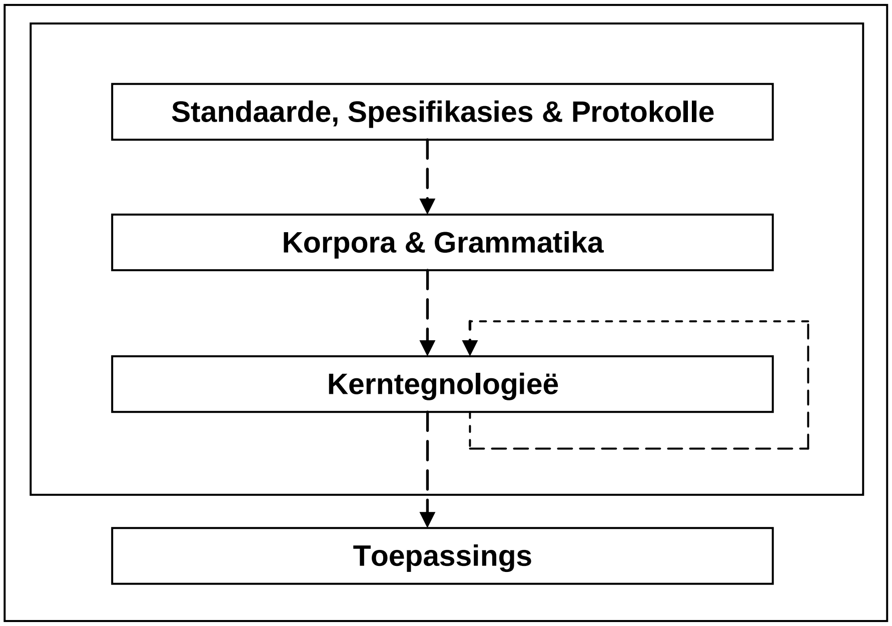
Hoofstuk 1

INLEIDING

1.1 AGTERGROND

Mensetaaltegnologie (Eng. *Human Language Technology*) is daardie tegnologie wat die interaksie tussen mens en rekenaar verbeter deur middel van natuurlike taal (Cole *et al.*, 1997). Dit word bewerkstellig deur die ontwikkeling van 'n verskeidenheid taaltegnologiese toepassings. Voorbeelde hiervan is onder andere teks-na-spraaksisteme, spel- en grammatika-toetsers, outomatiese vertaalsisteme, e-posfilters, ensovoorts. Die ideale stel taaltegnologiese toepassings sou kon bewerkstellig dat twee persone wat glad nie mekaar se taal verstaan nie, deur middel van 'n rekenaar met mekaar kan kommunikeer; of dat 'n blinde persoon toegang tot die internet sal hê; of dat 'n ongeletterde persoon toegang sal hê tot regeringsdokumente. In 'n ontwikkelende land soos Suid-Afrika, met elf amptelike landstale, sou gesê kon word dat die behoefte aan doeltreffende taaltegnologiese toepassings selfs nog groter is as in sommige Eerstewêreldlande met net een of twee amptelike landstale. Voordat enige taaltegnologiese toepassing vir 'n taal ontwikkel kan word, is dit noodsaaklik dat die nodige *hulpbronne* beskikbaar is.

Van Huyssteen (2003) verdeel hulpbronne in drie vlakke, waarvolgens sommige hulpbronne as hulpbronne vir 'n ander vlak hulpbronne en toepassings kan dien, soos wat in Figuur 1.1 geïllustreer word.



Figuur 1.1 – Hulpbronne as 'n voorvereiste vir die ontwikkeling van toepassings

Ten einde te verseker dat die tweede en derde vlak van hulpbronne funksioneel, betroubaar, bruikbaar, doeltreffend, onderhoubaar, oordraagbaar en herbruikbaar (Pilon, 2005) sal wees, word standaarde, spesifikasies en protokolle in die eerste vlak van hulpbronne vasgestel waaraan die opeenvolgende vlakke van hulpbronne moet voldoen. Die standaarde, spesifikasies en protokolle dien as riglyn om gesproke- en/of tekskorpora in te samel en te ontwikkel. Dit geld ook vir die ontwikkeling van grammatikas en kerntegnologieë.

’n *Kerntegnologie* (Eng. *enabling technology*) is hulpbronne wat op sigself reeds as ’n bruikbare toepassing gereken kan word. ’n Voorbeeld hiervan is ’n woordsoortetiketteerder wat dalk deur ’n woordeboekvervaardiger gebruik sal word as hulpmiddel om woorde te etiketteer. Woordsoortetikettering is die proses waartydens woordsoortetikette rekenaarmatig aan woorde toegeken word. ’n Woordsoortetiket bevat inligting rakende die woord waaraan dit toegeken is (Pilon, 2005). Dit is egter waarskynliker dat ’n woordsoortetiketteerder as bousteen binne ’n groter toepassing, soos ’n grammatikatoetser of teks-na-spraaksisteem, gebruik sal word. Van daar die benoeming kerntegnologie. Dit is ook moontlik vir een kerntegnologie om as hulpbron vir ’n ander kerntegnologie te dien. ’n Woordsoortetiketteerder word byvoorbeeld as hulpbron vir ’n sinstukverdeler (Eng. *chunker*), wat weer deel uitmaak van ’n grammatikatoetser gebruik. Kerntegnologieë word ook soms bemagtigingstegnologieë of modules genoem.

Die ontwikkeling van hulpbronne is ’n tydsame taak en dit vereis dikwels ’n hoë vlak van kundigheid (Dasgupta & Ng, 2007). Gevolglik is die ontwikkeling van hulpbronne in enige taal ’n duur proses en juis om hierdie rede bestaan daar baie tale, waaronder ook die inheemse tale van Suid-Afrika, wat nie oor voldoende taaltegnologiese hulpbronne beskik nie. Hierdie tale word gereken as sogenaamde *hulpbronskaars* tale (Eng. *resource-scarce languages*) (Peché, 2009). Sommige van die faktore wat die skaarsheid van taaltegnologiese hulpbronne vir ’n sekere taal beïnvloed sal vervolgens bespreek word:

- Aantal moedertaalsprekers: Vergelyk byvoorbeeld tale soos isiZulu en Afrikaans, wat onderskeidelik byna 12 miljoen en 7 miljoen moedertaalsprekers wêreldwyd het met Duits en Engels wat onderskeidelik 78.1 miljoen en meer as 335 miljoen moedertaalsprekers het. Boonop is Engels nog die mees gesproke nie-moedertaal in die wêreld (Lewis *et al.*, 2015);
- Politieke invloed: As ’n mens die getal Nederlandse moedertaalsprekers van 15.7 miljoen in Nederland en byna 22 miljoen wêreldwyd (Lewis *et al.*, 2015) met dié van isiZulu en Afrikaans vergelyk, sou ’n mens kon redeneer dat Nederlands ook amper as ’n hulpbronskaars taal gereken kan word. Nederlands word egter eerder aan die kant van tale soos Engels en Duits geskaar. Dit is waarskynlik te danke aan die politieke en finansiële ondersteuning wat die taal geniet. ’n Voorbeeld

hiervan is die STEVIN-projek wat ’n meerjarige ondersoek- en stimuleringsprogram vir Nederlandse teks- en spraaktegnologie is (Nederlandse Taalunie, 2011). Die projek het ’n begroting van 11.4 miljoen Euro wat gesamentlik befonds is deur die Vlaamse en Nederlandse owerhede; en

- Befondsing: Ten spyte van probleme soos armoede en ongeletterdheid besef die Suid-Afrikaanse regering die belangrikheid van taaltegnologiese toepassings in Suid-Afrika en befonds ’n projek[1] vir die ontwikkeling van herbruikbare digitale teks- en spraakhulpbronne vir tien van die Suid-Afrikaanse tale van Suid-Afrika.

Om hierdie redes moet die geld en tyd wat wel vir die ontwikkeling van taaltegnologiese toepassings beskikbaar gestel word so effektief as moontlik bestee word om die interaksie tussen mens en rekenaar deur middel van ’n persoon se moedertaal moontlik te maak.

Tien van die elf amptelike tale van Suid-Afrika, naamlik IsiZulu, IsiXhosa, Afrikaans, Sepedi, Setswana, Sesotho, Xitsonga, SiSwati, Tshivenda en IsiNdebele, kwalifiseer as hulpbronskaars tale.

Die hulpbronbestuursagentskap (RMA) (Eng. *Resource Management Agency*) vir die bestuur en verspreiding van herbruikbare digitale teks- en spraakhulpbronne vir al die amptelike tale van Suid-Afrika beskik oor verskeie tekshulpbronne wat wissel tussen korpusgrootte van 1.15 miljoen tot 3.27 miljoen tekseenhede (Eiselen & Puttkammer, 2014). Met woordsoortetiketteerderakkuraathede van tussen 82.08% en 96.02% (Eiselen & Puttkammer, 2014) vir die bogenoemde tien amptelike tale is die resultate vir die meeste Suid-Afrikaanse tale nog onder die vlak van die voorlopers in die wêreld, wat spog met akkuraathede van 96% tot 98% vir byvoorbeeld Nederlands, (Daelemans *et al.*, 1996, Poel *et al.*, 2007) en Engels (Brill, 1994, Garside & Smith, 1997).

Die laer akkuraatheid is te verwagte as in ag geneem word watter eerste vlak- en tweede vlakhulpbronne daar reeds vir die bogenoemde voorlopertale bestaan, vergeleke met die tien Suid-Afrikaanse tale. Om maar ’n paar te noem:

- Die Brown-korpus (Francis & Kucera, 1979) van een miljoen woorde is die pionier op die gebied van elektroniese tekskorpora en bestaan reeds vanaf 1964. Daar bestaan ’n ongeëtiketteerde weergawe en ’n weergawe wat met woordsoortetikette geëtiketteer is;
- Die Alpino-boombank van sintaktiese geëtiketteerde Nederlandse sinne bestaan uit meer as 150000 woorde (Bouma *et al.*, 2001);

[1] Projekhulpbronne beskikbaar by http://rma.nwu.ac.za/

- Die Lucy-korpus, wat ’n uitbreiding op die Susanne-korpus is, bestaan uit 165000 Engelse woorde wat in detail geëtiketteer is om grammatikale en ander linguistiese strukture te identifiseer (Sampson, 2003); en
- Die BNC (Eng. *British National Corpus*) XML Uitgawe (Consortium, 2007) bestaan uit 100 miljoen woorde, waarvan 90% teks is wat outomaties met woordsoort-etikette geëtiketteer is en 10% gesproke taal wat getranskribeer is (Burnard, 2007).

Sedert die negentiger jare het die vakgebied van taaltegnologie internasionaal baie aandag begin geniet en daar is verskeie toepassings ontwikkel en verbeteringe aangebring op onder andere die terreine van tekskategorisering, spraakherkenning, spraaksintese, inligting-onttrekking-sisteme en outomatiese vertaalsisteme. Op CoNLL-2000 het die beste sinstuk-verdeler ’n akkuraatheid van 93.48% behaal (Tjong Kim Sang & Buchholz, 2000). Op CoNLL-2001 (Daelemans & Zajac, 2001) was daar die gedeelde taak sinsnede-identifisering (Eng. *clause identification*), waar die beste sisteem ’n akkuraatheid van 78.63% behaal het (Sang & Déjean, 2001). Op CoNLL-X (Buchholz & Marsi, 2006) was die taak om ’n meertalige sins-ontleder (Eng. *multilingual dependency parser*) te ontwikkel wat vir enige taal afgerig kon word. Die deelnemers het vier dae tyd gehad om hierdie ontleder vir 13 tale af te rig. Die beste sins-ontleder het ’n herroeping van meer as 89% en ’n akkuraatheid van 91% behaal (Buchholz & Marsi, 2006).

Vir die inheemse tale van Suid-Afrika rapporteer Eiselen en Puttkammer (2014) akkuraathede van tussen 74.86% en 88.55% vir lemma-identifisering (Eng. *lemmatisation*), tussen 82.08% en 96.02% vir woordsoortetikettering en tussen 81.90% en 89.57% vir morfologiese ontbinding (Eng. *morphological decomposition*) deur gebruik te maak van die hulpbronne wat deur RMA beskikbaar gestel is. Taaltegnologiese toepassings vir alle Suid-Afrikaanse tale het nog nie tot op die vlak van meertalige sinsontleders gevorder nie. Verskillende redes hiervoor is hierbo bespreek.

Om hierdie agterstand te probeer verminder kan bestaande hulpbronne ondersoek word om maniere te probeer vind waarop dit effektiewer aangewend kan word. Dit sal moontlik lei tot die verbetering van die bestaande kerntegnologieë.

1.2 PROBLEEMSTELLING

Die fokus van hierdie studie is die etiketteringprobleem vir hulpbronskaars tale.

Tien van die elf amptelike tale van Suid-Afrika kwalifiseer as hulpbronskaars tale. Vir Afrikaans beskik RMA asook die Sentrum vir Tekstegnologie (CTexT®) oor verskeie hulpbronne wat verband hou met woordsoortetikettering.

Tegnieke en metodes vir die optimering van die hulpbronne en die verbetering in die akkuraatheid van die etiketteerder word in hierdie studie ondersoek en ontgin. Hierdie tegnieke en metodes moet sover moontlik nie taalspesifiek of toepassingspesifiek wees nie en moet, indien sukses daarmee behaal word vir Afrikaans, ook op ander hulpbronskaars tale en toepassings toegepas kan word.

Die akkuraatheid van die woordsoortetiketteerder kan moontlik verbeter word deur

- ontwikkeling van meer eerste en tweede vlak hulpbronne: Nog ’n manier waarop hulpbronne effektiewer aangewend kan word om akkuraatheid te verbeter is om meer as een woordsoortetiketteerder deur middel van enige kombinasie van masjienleeralgoritmes te ontwikkel en dan die woordsoortetiketteerders te kombineer in een kombinasie-woordsoortetiketteerder. Daar bestaan verskeie kombinasiemetodes, byvoorbeeld die kaskadekombinasiemetode (Eng. *cascading*) (Karahan *et al.*, 2003), die stemmetode (Karahan *et al.*, 2003; Van Halteren *et al.*, 2001) en die opstapelingmetode (Eng. *stacking*) (Karahan *et al.*, 2003; Van Halteren *et al.*, 2001), om maar ’n paar te noem; en
- reeds bestaande hulpbronne aan te wend deur ’n woordsoortetiketteerder af te rig deur middel van gekontroleerde, gedeeltelik-gekontroleerde, of ongekontroleerde masjienleeralgoritmes. Die woordsoortetiketteerder wat deur Pilon (2005) ontwikkel is, maak gebruik van TnT, wat ’n gekontroleerde masjienleeralgoritme is. Die eerste en tweede vlak hulpbronne wat deur Pilon (2005) gebruik is, sluit in ’n woordsoortetiketstel en ’n stel korrekgeëtiketteerde tekseenhede, waarvan ongeveer 10% as toetsdata gebruik is. Gekontroleerde masjienleeralgoritmes maak egter nie voorsiening vir die gebruik van rou elektroniese teks, wat in hierdie geval ’n belangrike beskikbare hulpbron is, nie. Aan die ander kant maak ongekontroleerde masjienleeralgoritmes weer slegs voorsiening vir die gebruik van rou elektroniese teks, en word die korrekgeëtiketteerde tekseenhede nie in berekening gebring nie. Gedeeltelik-gekontroleerde leer maak gebruik van beide geëtiketteerde data en ’n groot hoeveelheid ongeëtiketteerde data (rou elektroniese teks) om beter klassifiseerders te bou (Zhu, 2005).

In hierdie studie is die fokus op gedeeltelik-gekontroleerde masjienleeralgoritmes, aangesien dit voorsiening maak vir die gebruik van soveel as moontlik van die beskikbare hulpbronne. Gedeeltelik-gekontroleerde leer sal dus net soos in Pilon (2005) se gekontroleerde leer gebruik maak van ’n woordsoortetiketstel, maar addisioneel ook gebruik maak van die rou elektroniese teks, wat akkuraatheid behoort te verbeter.

1.3 NAVORSINGSVRAE

Met die oog op die verbetering van ’n Afrikaanse woordsoortetiketteerder met behulp van die beskikbare hulpbronne word die volgende vrae gestel:

1. Kan kombinasietegnieke gebruik word om die akkuraatheid van ’n Afrikaanse woordsoortetiketteerder te verbeter?
2. Kan strukturele gedeeltelik-gekontroleerde leer effektief aangewend word om die akkuraatheid van ’n gekontroleerde leer Afrikaanse woordsoortetiketteerder te verbeter?

1.4 DOELSTELLINGS

Om die navorsingsvrae te beantwoord word die volgende doelstellings daargestel.

1. Om te bepaal of kombinasietegnieke effektief gebruik kan word om die akkuraatheid van ’n Afrikaanse woordsoortetiketteerder te verbeter, word vier woordsoortetiketteerders wat deur middel van verskillende masjienleeralgoritmes ontwikkel is in ’n kombinasie-woordsoortetiketteerder saamgevoeg.
2. Om te bepaal of strukturele gedeeltelik-gekontroleerde leer effektief is, word ’n algoritme geïmplementeer wat op gedeeltelik-gekontroleerde leer gebaseer is en word dit gebruik om ’n kerntegnologie, in hierdie geval ’n woordsoortetiketteerder vir Afrikaans, af te rig.

1.5 SENTRALE TEORETIESE STELLING

Volgens Dietterich (1997), Hoste en Daelemans (2000) en Utsuro *et al.* (2002), sal ’n kombinasieklassifiseerder beter vaar as enige van die klassifiseerders waaruit dit saamgestel is indien die korrelasie tussen die foute van die onderliggende klassifiseerders klein genoeg is en die persentasie foute wat hulle begaan laag genoeg is.

Ando en Zhang (2005a), Ando en Zhang (2005b) en Zhu (2009) is dit eens dat ongeëtiketteerde data nie noodwendig die akkuraatheid van ’n kerntegnologie of toepassing sal verbeter nie. So byvoorbeeld het deelleer (Eng. *co-training*) en selfleer (Eng. *self-training*) in die eksperimente

van Ando en Zhang (2005a) en Ando en Zhang (2005b) die herroeping van ’n sintaktiese sinstukverdeler effens verbeter, maar die akkuraatheid weer effens laat daal. Om hierdie rede is dit baie belangrik dat die regte gedeeltelik-gekontroleerde masjienleeralgoritme gekies word, sodat ongeëtiketteerde data wél nuttig is en dit die akkuraatheid sal verbeter. In hierdie studie word ’n metode gebruik wat reeds met sukses op ander tale toegepas is (Ando & Zhang, 2005a; Ando & Zhang, 2005b) om hopelik ook sukses vir Afrikaans te behaal. Die metode is gebaseer op strukturele leer (Eng. *structural learning*) en gebruik ’n SVD-ASO-algoritme (Eng. *Singular Value Decomposition – Alternating Structure Optimisation – Algorithm*) om nuttige inligting uit ongeëtiketteerde data te leer. Hierdie inligting wat uit die ongeëtiketteerde data geleer is, word as eienskapvektore saam met die eienskapvektore wat uit die geëtiketteerde data geleer is, gekombineer in ’n klassifiseerder.

Ando en Zhang (2005a) en Ando en Zhang (2005b) se SVD-ASO-sintaktiese sinstukverdeler vaar beter as die wensisteem vir die CoNLL 2000 gedeelde taak (Tjong Kim Sang & Buchholz, 2000), met ’n akkuraatheid van 94.39% teenoor die 93.48% van Kudo en Matsumoto (2001), wie se sisteem ’n kombinasie van steunvektorklassifiseerders is en slegs geëtiketteerde data gebruik; die SVD-ASO-sisteem maak addisioneel van ongeëtiketteerde data gebruik. Ando en Zhang (2005a) en Ando en Zhang (2005b) se SVD-ASO-benoemde entiteitherkenningsisteem vaar ook beter as die CoNLL 2003-wenner (Daelemans & Osborne, 2003) met ’n akkuraatheid van 94.39% teenoor 88.76% vir Engels, en 75.27% teenoor 72.41% vir Duits.

Op grond van hierdie resultate is dit te verwagte dat Ando en Zhang (2005a) en Ando en Zhang (2005b) se gedeeltelik-gekontroleerde leermetode ook vir ’n Afrikaanse woordsoortetiketteerder goeie resultate sal lewer.

1.6 NAVORSINGSMETODES

Om die bogenoemde doelstellings te bereik, word die volgende navorsingsaktiwiteite in hierdie studie uitgevoer:

1.6.1 LITERATUURSTUDIE

Gedurende die fundamentele navorsingsfase word ’n literatuurondersoek oor die volgende onderwerpe gedoen:

- Masjienleer, meer spesifiek gekontroleerde leer wat klassifikasie behels, sowel as ’n verskeidenheid bestaande gedeeltelik-gekontroleerde leeralgoritmes;
- Klassifikasiekombinasiemetodes: Volgens Dietterich (1997), Hoste en Daelemans (2000) en Utsuro *et al.* (2002), sal ’n kombinasieklassifiseerder beter vaar as enige

van die klassifiseerders waaruit dit saamgestel is indien die korrelasie tussen die foute van die onderliggende klassifiseerders klein genoeg is en die persentasie foute wat hulle begaan laag genoeg is; en

- Strukturele gedeeltelik-gekontroleerde leer: Die metode wat reeds met sukses op ander tale toegepas is (Ando & Zhang, 2005a; Ando & Zhang, 2005b), is gebaseer op strukturele leer en gebruik ’n SVD-ASO-algoritme om nuttige inligting uit ongeëtiketteerde data te leer. Hierdie inligting wat uit die ongeëtiketteerde data geleer is, word as eienskapvektore saam met die eienskapvektore wat uit die geëtiketteerde data geleer is, gekombineer in ’n klassifiseerder.

Die doel van die literatuurondersoek is om die belangrikste kombinasiemetodes wat reeds bestaan en goeie resultate opgelewer het te ondersoek, waarna ’n metode gekies word om tydens toegepaste navorsing te implementeer. Verder is die doel om ’n omvattende begrip van gedeeltelik-gekontroleerde leer deur middel van strukturele leer binne die konteks van masjienleer te verkry, sodat die metode in die toegepaste navorsing geïmplementeer kan word.

1.6.2 EKSPERIMENTERING

Die volgende word geïmplementeer:

1. ’n Kombinasietegniek vir ’n kombinasie-woordsoortetiketteerder deur vier gekontroleerde woordsoortetiketteerders te kombineer; en
2. Die SVD-ASO-algoritme (Ando & Zhang, 2005a) om ’n gedeeltelik-gekontroleerde Afrikaanse woordsoortetiketteerder af te rig.

’n Reeks evaluasiemetrieke wat van toepassing is op hierdie studie en wat algemeen spesifiek vir woordsoortetikettering gebruik word, (Ando & Zhang, 2005a; Van Halteren *et al.*, 2001) word vervolgens kortliks verduidelik.

1.6.3 PRESISIE, HERROEPING EN AKKURAATHEID

Twee welbekende maatstawwe vir klassifisering is *presisie* (Eng. *precision*) en *herroeping* (Eng. *recall*). Vir die binêre besluitnemingsprobleem, wat van toepassing is in hierdie studie, word die klassifiseerder gebruik om voorbeelde uit die afrigtingsdata as positief of negatief te etiketteer. Die keuse wat deur die klassifiseerder uitgeoefen word val in een van vier kategorieë naamlik werklike positiewe (TP) (Eng. *True Positives*), vals positiewe (FP) (Eng. *False Positives*), werklike negatiewe (TN) (Eng. *True Negatives*) of vals negatiewe (FN) (Eng. *False Negatives*). TP is voorbeelde wat korrek as positief geëtiketteer is, terwyl FP verwys na negatiewe voorbeelde wat verkeerdelik as positief geëtiketteer is. TN verwys na negatiewe voorbeelde wat

korrek as negatief geëtiketteer is en FN is positiewe voorbeelde wat verkeerdelik as negatief geëtiketteer is. Die verwarringsmatriks in Tabel 1.1 gee ’n voorstelling van die vier kategorieë soos hierbo verduidelik.

	Werklike positief	Werklike negatief
Voorspelde positief	TP	FP
Voorspelde negatief	FN	TN

Tabel 1.1 – Verwarringsmatriks vir werklike en voorspelde klassifikasies

Gegee die verwarringsmatriks in Tabel 1.1 word presisie en herroeping in (1.1) en (1.2) gedefinieer.

$$Presisie = \frac{TP}{TP + FP} \tag{1.1}$$

$$Herroeping = \frac{TP}{TP + FN} \tag{1.2}$$

Vir enige etiket X meet presisie die persentasie tekseenhede waaraan die etiket X toegeken is wat wel etiket X in die toetsdata het, terwyl herroeping meet watter persentasie tekseenhede met etiket X in die toetsdata ook deur die klassifiseerder met etiket X toegeken is (Davis & Goadrich, 2006; Van Halteren *et al.*, 2001).

Akkuraatheid meet hoeveel klassifikasiegevalle in totaal vir ’n klassifiseerder korrek geklassifiseer is, aldus die totale presisie. In hierdie geval is presisie en herroeping gelyk vir klassifiseerders wat net een etiket per klassifikasiegeval toeken (Van Halteren *et al.*, 2001).

1.6.4 FOUTKOERS EN FOUTKOERSREDUKSIE

Foutkoers (Eng. *error rate*) is ’n alternatiewe manier om akkuraatheid uit te druk, naamlik $1 - Akkuraatheid$. *Foutkoersreduksie* (Eng. *error rate reduction*) is die vermindering in foutkoers met betrekking tot ’n basis (Van Halteren *et al.*, 2001) en dui aan hoeveel akkurater ’n klassifiseerder as ’n basislynklassifiseerder is.

1.6.5 MCNEMAR SE TOETS

McNemar se toets (Dietterich, 1998) word gebruik om statistiese betekenisvolheid te bepaal. Dit behels dat Klassifiseerder A en Klassifiseerder B afgerig word op dieselfde afrigtingsdata en daarna op dieselfde toetsdata T getoets word. Die resultaat van elke klassifikasiegeval in T wat deur Klassifiseerder A en Klassifiseerder B geklassifiseer is, word volgens die verwarrings-matriks in Tabel 1.2 met notasie in (1.3) getel. Vergelyking (1.4) is die totale aantal klassifikasie-gevalle in T.

Aantal voorbeelde wat verkeerd geklassifiseer is deur Klassifiseerder A en Klassifiseerder B.	Aantal voorbeelde wat verkeerd geklassifiseer is deur Klassifiseerder A en reg geklassifiseer is deur Klassifiseerder B.
Aantal voorbeelde wat verkeerd geklassifiseer is deur Klassifiseerder B en reg geklassifiseer is deur Klassifiseerder A.	Aantal voorbeelde wat reg geklassifiseer is deur Klassifiseerder A en Klassifiseerder B.

Tabel 1.2 – Verwarringsmatriks vir klassifikasies van Klassifiseerder A en Klassifiseerder B

n_{00}	n_{01}
n_{10}	n_{11}

(1.3)

$$n = n_{00} + n_{01} + n_{10} + n_{11} \tag{1.4}$$

Die statistiek wat in (1.5) gegee word (voortaan McNemar-statistiek) is benaderd verdeel as χ^2 met 1 vryheidsgraad. Die waarskynlikheid dat hierdie waarde groter is as 3.841459 is dus kleiner as 0.05 en die hipotese dat die twee klassifiseerders dieselfde foutkoers het, kan in daardie geval verwerp word.

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{1.5}$$

1.7 ONTPLOOIING VAN NAVORSING

Hoofstuk 2 bring gedeeltelik-gekontroleerde leer en klassifikasie, asook kombinasiemetodes in verband. Klassifikasie met twee bekende algoritmes word binne die konteks van gedeeltelik-gekontroleerde masjienleer bespreek waarna verskeie gedeeltelik-gekontroleerde masjienleer-algoritmes bespreek word. In die laaste afdeling word die belangrikste kombinasiemetodes bespreek.

In Hoofstuk 3 word vier verskillende gekontroleerde leeralgoritmes beskryf waarna ’n beskrywing van die eksperimente om die vier algoritmes te kombineer volg. Die resultate van die vier woordsoortetiketteerders, asook die kombinasie-woordsoortetiketteerder word bespreek.

Hoofstuk 4 begin met ’n beskrywing van strukturele leer en ’n wisselstruktuur-optimeringsalgoritme. In die eksperimentele opstelling word die generering van hulpbronne en ’n gedeelde struktuur uit die ongeëtiketteerde data beskryf. Verder volg ’n beskrywing van hoe die gedeelde struktuur aangewend word in die afrigting van ’n Afrikaanse woordsoort-etiketteerder. Die hoofstuk sluit af met ’n bespreking van die resultate.

In Hoofstuk 5 word ’n samevatting van die studie gegee en gevolgtrekkings word gemaak oor die kombinasie van masjienleeralgoritmes en die toepassing van strukturele gedeeltelik-gekontroleerde leer vir woordsoortetikettering vir hulpbronskaars tale. Toekomstige navorsings-moontlikhede en -onderwerpe word dan ter afsluiting van die studie genoem.

Hoofstuk 2

INLEIDING TOT LEER- EN KOMBINASIEMETODES

Hierdie hoofstuk bespreek oorsigtelik gekontroleerde, ongekontroleerde en gedeeltelik-gekontroleerde leer met ’n basiese beskrywing van ’n paar masjienleeralgoritmes wat in die literatuur bestaan. Verder word verskillende tegnieke vir die kombinering van masjienleer-algoritmes bestudeer.

2.1 MASJIENLEER

In 1959 ontwikkel Arthur Samuel (McCarthy & Feigenbaum, 1990) ’n dambordprogram wat illustreer hoe masjienleer dit moontlik maak vir ’n rekenaar om ’n vaardigheid aan te léér eerder as wat dit eksplisiet geprogrammeer hoef te word (Samuel, 2000). Hy programmeer sy program om dambord herhaaldelik teen homself te speel, sodat dit met tyd leer om tussen patrone wat lei tot ’n wen en patrone wat lei tot ’n verloor te onderskei. So slaag Samuel nie net daarin om ’n program te ontwikkel wat geleer het hoe om dambord te speel nie, maar hy slaag ook daarin om ’n dambordprogram te ontwikkel wat selfs beter speel as Samuel self.

Soos in die geval van Samuel se dambordprogram maak masjienleer dit vir rekenaars moontlik om te leer sonder om die taak ter sprake eksplisiet te programmeer.

Masjienleer (Schapire, 2003a) bestudeer rekenaaralgoritmes om spesifieke take aan te leer deur onder meer patrone in die data te ontdek en daarvolgens die program aan te pas. Die leer wat plaasvind is altyd gebaseer op die waarneming van data, soos voorbeelde, direkte ondervinding of instruksie. Die klem van masjienleer is op outomatiese metodes, dus om leer-algoritmes af te lei wat outomaties sal leer met so min as moontlik inmenging deur die mens. Die paradigma van masjienleer kan gesien word as programmering deur middel van voorbeelde.

> Formeler gestel: ’n rekenaarprogram leer uit ervaring E met verwysing na ’n klas van take T en ’n prestasiemeting P, indien die prestasie op take T, soos gemeet deur P, verbeter met ervaring E (Mitchell, 1997).

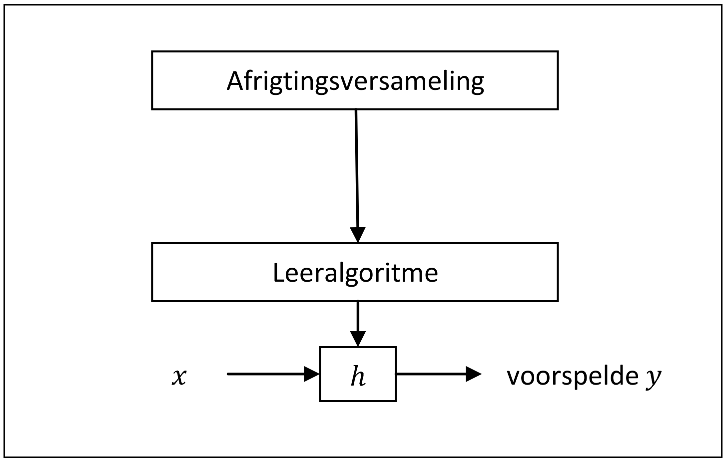
Masjienleer is ’n kernonderafdeling van kunsmatige intelligensie (Schapire, 2003a), alhoewel daar ook sterk ooreenkomste is met ander vakgebiede waaronder spesifiek Statistiek.

Masjienleer kan plaasvind met behulp van voorbeelde wat lyk soos die teikenprobleem of daar kan geen voorbeelde nie maar wel een of ander metode of hulpbron waaruit die masjienleer-algoritme kan leer, beskikbaar wees, of dit kan ’n kombinasie van die twee wees. Wanneer voorbeelde beskikbaar is, word dit gekontroleerde leer genoem en wanneer geen eksplisiete

voorbeelde bestaan nie word dit ongekontroleerde leer genoem. ’n Kombinasie van die twee word gedeeltelik-gekontroleerde leer genoem. Vervolgens word hierdie drie beskrywings van leer bespreek.

2.1.1 GEKONTROLEERDE LEER

Gegee daar bestaan ’n versameling afrigtingsdata wat gebruik word vir die leer van ’n funksie $h : X \rightarrow Y$ sodat $h(x)$ ’n “goeie” voorspeller vir die ooreenstemmende waarde van y is (Ng, 2000). Die proses word voorgestel in Figuur 2.1.



Figuur 2.1 – Die gekontroleerde leerprobleem

Wanneer die voorspelde teikenwaarde kontinu is, word die leerprobleem as ’n regressie-probleem beskryf, maar wanneer y slegs ’n klein hoeveelheid diskrete waardes kan aanneem, word die leerprobleem as ’n klassifikasieprobleem beskryf (Ng, 2000). Vir die doel van hierdie studie is die fokus op klassifikasie.

2.1.1.1 KLASSIFIKASIE

In die meeste toepassings van masjienleersisteme op die gebied van natuurliketaalverwerking word die natuurliketaalverwerkingstaak omgewerk na ’n klassifikasietaak (Van Halteren, 1999). In ’n *klassifikasietaak* ontvang ’n *klassifiseerder* invoer in die vorm van ’n *eienskapvektor* en gee ’n *klassifikasie* as uitvoer wat deur middel van ’n bepaalde metode uit ’n eindige lys moontlike klassifikasies gekies is. ’n Eienskapvektor kan *atomies* of *kompleks* van aard wees (Van Halteren, 1999). ’n Atomiese eienskapvektor is ’n eienskapvektor wat uit een enkele eienskap bestaan – dit kan nie opgedeel word in kleiner, eenvoudiger eienskapvektore nie, terwyl ’n komplekse eienskapvektor bestaan uit ’n aantal atomiese eienskapvektore. Die eienskapvektor as invoer en die klassifikasie as uitvoer word saam ’n *klassifikasiegeval* genoem. Voordat die klassifiseerder ’n klassifikasie toegeken het, bestaan die klassifikasiegeval uit die

eienskapvektor en ŉ onbekende klassifikasie (Van Halteren, 1999). Daar word dus steeds van die klassifikasiegeval gepraat voordat klassifikasie plaasgevind het.

ŉ Gekontroleerde leer klassifikasie-algoritme word geïmplementeer om ŉ klassifiseerder te genereer (Elkan, 2011). Die algoritme op sigself is nie ŉ klassifiseerder nie, maar wel die uitset daarvan. Vervolgens word twee konkrete voorbeelde van klassifikasie-algoritmes wat algemeen in die literatuur voorkom, bespreek.

Die naïewe Bayes-algoritme

Die naïewe Bayes-algoritme (Abney, 2007) is eenvoudig, maar vaar tog bo verwagting goed gegewe die “naïewe” aanname wat gemaak word.

Gegee ŉ komplekse eienskapvektor wat bestaan uit ŉ aantal atomiese eienskapvektore: die komplekse eienskapvektor kan voorgestel word as ŉ vektor $\overline{x}$ wat uit ŉ aantal attribute (atomiese eienskapvektore) bestaan, soos in (2.1).

$$\overline{x} = (x_1, x_2, \dots x_m) \tag{2.1}$$

Die kettingreël van voorwaardelike waarskynlikheid bepaal dat die waarskynlikheid dat $\overline{x}$ ŉ klassifikasie y het, gelyk is aan die waarskynlikheid dat y voorkom vermeningvuldig met die waarskynlikheid dat x_i voorkom, gegee dat y en x_{i-1} ... x_1 waar is, soos voorgestel in (2.2).

$$p(\overline{x}, y) = p(y)p(x_1|y) \dots p(x_m|y, x_1, .., x_i, \dots, x_{m-1}) \tag{2.2}$$

Die naïwiteit van die algoritme lê in ŉ onafhanklikheidsaanname, naamlik dat die keuse van die waarde van enige attribuut voorwaardelik onafhanklik is van die keuse van die waarde van enige ander attribuut, gegewe die klassifikasie, soos voorgestel in (2.3).

$$p(\overline{x}, y) = p(y) \prod_j p(x_j|y) \tag{2.3}$$

Wanneer ŉ klassifiseerder dus moet bepaal watter klassifikasie aan ŉ eienskapvektor toegeken word, kies dit die klassifikasie waar die waarskynlikheid van klassifikasie y die grootste is, soos voorgestel in (2.4).

$$\underset{y}{arg\,max}\ p(y) \prod_j p(x_j|y) \tag{2.4}$$

Vervolgens word die naïewe Bayes-algoritme soos verduidelik hierbo op ŉ baie eenvoudige voorbeeld toegepas. Afrikaanse woordsoortetikettering (Pilon, 2005) word as klassifikasietaak gebruik, met ŉ woordsoortetiketteerder as klassifiseerder. Gestel die woordsoortetiketteerder moet die woord "ander" in die sin "Hy het my ’n ander kind geskenk." etiketteer. Die eienskapvektore, soos uitgebeeld in (2.5) is beskikbaar.

$$\begin{aligned} \overline{x} &= (x_1, x_2, x_3)\ ,\ waar \\ x_1 &= \{Tekseenheid = ander\} \\ x_2 &= \{Vorigetekseenheid = {}'n\} \\ x_3 &= \{Volgendetekseenheid = kind\} \end{aligned} \tag{2.5}$$

Neem aan dat die woordsoortetiketteerder tydens die afrigtingsfase ’n aantal komplekse eienskapvektore $\overline{a}$ tot $\overline{k}$ ontvang:

$$\overline{a} = (\{Tekseenheid = \text{rustige}\}, \{Vorigetekseenheid = \text{die}\}, \{Volgendetekseenheid = \text{kind}\})$$

.

.

.

$$\overline{k} = (\{Tekseenheid = \text{een}\}, \{Vorigetekseenheid = \text{net}\}, \{Volgendetekseenheid = \text{kind}\})$$

Die eienskapvektore met klassifikasies word in Tabel 2.1 uiteengesit.

	Tekseenheid	*Vorigetekseenheid*	*Volgendetekseenheid*	Klassifikasie $\boldsymbol{y}$ (vanuit afrigtingsdata)
$\bar{a}$	rustige	die	kind	ASA
$\bar{b}$	koue	’n	winterluggie	ASA
$\bar{c}$	mooi	die	blom	ASA
$\bar{d}$	mooi	hardloop	.	BSW
$\bar{e}$	in	is	die	SV
$\bar{f}$	ander	’n	program	THAB
$\bar{g}$	ander	na	mense	ASA
$\bar{h}$	ander	by	universiteite	ASA
$\bar{i}$	ander	net	lede	ASA
$\bar{j}$	ander	van	en	NSE
$\bar{k}$	een	net	kind	THAB

Tabel 2.1 – Komplekse eienskapvektore

In Tabel 2.2 word ’n naïewe Bayes-model vanuit die afrigtingsdata gebou (Haruechaiyasak, 2008) om die waarskynlikheid van “ASA” (Pilon, 2005), wat ’n adjektief is, as woordsoortetiket te bereken (kyk Bylaag B vir volledige beskrywings van etikette). Die boonste gedeelte van Tabel 2.2 bevat die frekwensies van die verskillende atomiese eienskapvektore vir die moontlike uitkomste van “ASA”. Daar is byvoorbeeld 2 voorbeelde vanuit die afrigtingsdata waar $(Vorigetekseenheid = \text{die})$ wanneer $(y = ASA)$. In die onderste gedeelte van Tabel 2.2 word die naïewe Bayes-model gebou deur alle $p(\mathrm{x_i}|\mathrm{y})$ en $p(\mathrm{y})$ te bereken, byvoorbeeld $p(Tekseenheid = \text{ander}|y = \text{ASA}) = 3/6$ en $p(y = \text{ASA}) = 6/11$

Tekseenheid			*Vorigetekseenheid*			*Volgendetekseenheid*			ASA	
	ASA	!ASA		ASA	!ASA		ASA	!ASA	ASA	!ASA
rustige	1	0	die	2	0	kind	1	1	6	5
koue	1	0	’n	1	1	winterluggie	1	0		
mooi	1	1	hardloop	0	1	blom	1	0		
in	0	1	is	0	1	.	0	1		
ander	3	2	na	1	0	die	0	1		
een	0	1	by	1	0	program	0	1		
			net	1	1	mense	1	0		
			van	0	1	universiteite	1	0		
						lede	1	0		
						en	0	1		
rustige	1/6	0/5	die	2/6	0/5	kind	1/6	1/5	6/11	5/11
koue	1/6	0/5	’n	1/6	1/5	winterluggie	1/6	0/5		
mooi	1/6	1/5	hardloop	0/6	1/5	blom	1/6	0/5		
in	0/6	1/5	is	0/6	1/5	.	0/6	1/5		
ander	3/6	2/5	na	1/6	0/5	die	0/6	1/5		
een	0/6	1/5	by	1/6	0/5	program	0/6	1/5		
			net	1/6	1/5	mense	1/6	0/5		
			van	0/6	1/5	universiteite	1/6	0/5		
						lede	1/6	0/5		
						en	0/6	1/5		

Tabel 2.2 – Die naïewe Bayes-model op woordsoortetiket: ASA

Vervolgens kan die model in Tabel 2.2 gebruik word om die waarskynlikheid vir die woordsoortetiket “ASA” vir ’n gegewe klassifikasiegeval te bereken. Tabel 2.3 gee die beskikbare eienskapvektore vir die klassifisering van “ander”.

	Tekseenheid	*Vorigetekseenheid*	*Volgendetekseenheid*	Klassifikasie y
$\overline{x}$	ander	’n	kind	?

Tabel 2.3 – Klassifikasiegeval

Die waarskynlikheid dat “ander” die woordsoortetiket “ASA” het, word in (2.6) bereken. Die waarskynlikheid dat die woordsoortetiket “ASA” nie die klassifikasie vir “ander” is nie, word in (2.7) bereken.

$$
\begin{aligned}
p(ASA|\overline{x}) = \frac{p(\overline{x}, ASA)}{p(\overline{x})} &= p(x_1|ASA)p(x_2|ASA)p(x_3|ASA)p(ASA) \times \frac{1}{p(\overline{x})} \\
&= p(Tekseenheid = ander|ASA) \\
&\quad \times p(Vorigetekseenheid = \text{'}n|ASA) \\
&\quad \times p(Volgendetekseenheid = kind|ASA) \\
&\quad \times p(ASA) \\
&\quad \times \frac{1}{p(\overline{x})} \\
&= \frac{3}{6} \times \frac{1}{6} \times \frac{1}{6} \times \frac{6}{11} \times \frac{1}{p(\overline{x})} \\
&= \frac{0.00757576}{p(\overline{x})}
\end{aligned} \tag{2.6}
$$

$$
\begin{aligned}
!ASA|\overline{x}) = \frac{p(\overline{x}, !ASA)}{p(\overline{x})} &= p(x_1|!ASA)p(x_2|!ASA)p(x_3|!ASA)p(!ASA) \times \frac{1}{p(\overline{x})} \\
&= p(Tekseenheid = ander|!ASA) \\
&\quad \times p(Vorigetekseenheid = \text{'}n|!ASA) \\
&\quad \times p(Volgendetekseenheid = kind|!ASA) \\
&\quad \times p(!ASA) \\
&\quad \times \frac{1}{p(\overline{x})} \\
&= \frac{2}{5} \times \frac{1}{5} \times \frac{1}{5} \times \frac{5}{11} \times \frac{1}{p(\overline{x})} \\
&= \frac{0.00727273}{p(\overline{x})}
\end{aligned} \tag{2.7}
$$

Die waarskynlikheid dat die woord “ander” die adjektief-woortsoortetiket “ASA” het, is dus groter as die waarskynlikheid dat dit ’n ander klassifikasie het. Die Woordsoortetiketteerder ken in hierdie vereenvoudigde geval “ASA” aan $\overline{x}$ toe.

k-Naastebuurpuntalgoritme

Die *k*-naastebuurpuntalgoritme (Abney, 2007; Elkan, 2011; Mitchell, 1997) is die teenoorgestelde van die naïewe Bayes-algoritme in die sin dat dit gedurende die afrigtingsfase alle klassifikasiegevalle wat in die afrigtingsdata gegee is, memoriseer en die werklike verwerking eers tydens klassifikasie plaasvind.

Gegee eienskapvektor $\overline{x}$ word as klassifikasietaak voorgelê, dan sal die klassifiseerder die *k*-aantal klassifikasiegevalle uit die afrigtingsdata neem waar die eienskapvektore die meeste ooreenstem met $\overline{x}$ en hul klassifikasies vergelyk.

Om die *k*-naastebuurpuntalgoritme toe te pas word daar eerstens ŉ waarde vir k benodig en tweedens ŉ metode om te bepaal watter eienskapvektore die meeste ooreenstem met die klassifikasietaak, of anders gestel, watter k klassifikasiegevalle die naaste aan $\bar{x}$ lê, dus ŉ metode om die *eendersheid* te bepaal.

k word toegeken sodat $k \geq 1$ en k ŉ heelgetal is. Indien $k > 1$ kan daar ŉ stemming tussen die *k*-naaste omliggende punte plaasvind om te bepaal watter klassifikasie aan $\bar{x}$ toegeken word.

Dit is verkieslik dat k as ŉ onewe getal gekies word, aangesien dit die kans om 'n dooie punt te bereik tydens 'n stemming tussen die *k*-naaste omliggende punte verminder. ŉ Eenvoudige geometriese interpretasie van die algoritme is om die eendersheid deur middel van die Euklidiese afstand te meet. Die Euklidiese afstand na elke klassifikasiegeval in die afrigtingsdata word bereken, waarna slegs die *k*-naaste klassifikasiegevalle gehou word. Die klassifikasie wat die meeste tussen die voorbeelde voorkom word as klassifikasie aan die eienskapvektor toegeken. Die Euklidiese afstand word in (2.8) getoon.

$$(x, y) = \|x - y\| = \sqrt{(x - y).(x - y)} = \left(\sum_{i=1}^{m} (x_i - y_i)^2\right)^{\frac{1}{2}} \text{ waar } x \text{ en } y \text{ punte in } X = \mathbb{R}^m \tag{2.8}$$

Die *k*-naastebuurpuntalgoritme genereer komplekse modelle tydens die afrigtingsfase omdat dit nie die data verwerk nie, maar eerder elke klassifikasiegeval stoor wat 'n goeie passing tot gevolg het. Die teenpool hiervan is dat die algoritme nie effektief ruis in die afrigtingsdata kan hanteer nie.

Hierdie afdeling het twee algemene algoritmes vir gekontroleerde leer bespreek. Die volgende afdeling bevat 'n inleidende bespreking van ongekontroleerde leer.

2.1.2 ONGEKONTROLEERDE LEER

Ongekontroleerde leermetodes behels die voorstelling van spesifieke patrone sodat dit die statistiese struktuur van die algehele versameling van invoerpatrone reflekteer. In teenstelling met gekontroleerde leer bestaan daar geen teikenuitsette wat met die invoer geassosieer word nie (Dayan, 1999). Met ander woorde, die afrigtingsdata waaruit geleer moet word, is ongeëtiketteer en voorbeelde van volledige klassifikasiegevalle word nie verskaf nie, dus moet etikette uit die data afgelei word (Klein & Manning, 2002). Die voordeel is dat ongeëtiketteerde afrigtingsdata makliker bekombaar is as geëtiketteerde afrigtingsdata en nie menige arbeids-intensiewe ure deur kundiges vereis om te skep nie. In die ongekontroleerde leeropset is alle afrigtingsdata $\bar{X} = \{\bar{x}_i\}_{i=1}^{N}$ ongeëtiketteer, waar N die aantal afrigtingspatrone voorstel (Huang *et al.*, 2014). Die teiken is om die onderliggende struktuur van die oorspronklike data te vind. Twee

klassieke voorbeelde van ongekontroleerde leer is *trosvorming* (Eng. *clustering*) en *dimensionaliteitvermindering* (Eng. *dimensionality reduction*) (Ghahramani, 2004).

Trosvorming (Grira *et al.*, 2004) word in die ongekontroleerde domein gebruik wanneer geen inligting oor die klasse waaraan die data behoort beskikbaar is nie en het ten doel om 'n data-versameling van items in trosse te organiseer sodat items in elke tros meer soortgelyk aan mekaar sal wees as aan items in ander trosse.

Modelle wat dimensionaliteitvermindering (Fodor, 2002; Rajaraman & Ullman, 2011) toepas word op hoëdimensionele datastelle gebruik (Ghahramani, 2004). Data word dikwels as 'n groot matriks voorgestel en menigmaal is dit ook noodsaaklik om so 'n groot matriks op te som in een of meer kleiner matrikse met 'n klein aantal rye of kolomme wat op 'n manier tog soortgelyk aan die oorspronklike is. Hierdie kleiner matrikse kan rekenaarmatig effektiewer gebruik word en die proses om hierdie kleiner matrikse te skep word dimensionaliteitvermindering genoem (Rajaraman & Ullman, 2011).

Modelle wat gebruik word in trosvorming en dimensionaliteitvermindering sluit in faktoranalise (Eng. *factor analysis*), hoofkomponente-analise (Eng. *principal components analysis (PCA)*), onafhanklike komponente-analise (Eng. *Independent components analysis (ICA*)), Mengsel van Gausse (Eng. *Mixture of Gaussians*) en *K*-gemiddeldes (Eng. *K-means*) (Ghahramani, 2004).

Die volgende afdeling bespreek algoritmes wat nie ten volle op ongekontroleerde leer gebaseer is nie en wel ook 'n gekontroleerde deel bevat, naamlik gedeeltelik-gekontroleerde leer.

2.1.3 Gedeeltelik-gekontroleerde leer

In die vorige twee afdelings is gekontroleerde en ongekontroleerde leer bespreek. 'n Logiese vraag wat by die leser sou kon opkom, is of dit moontlik is om laasgenoemde twee metodes te kombineer. Gedeeltelik-gekontroleerde leer verwys juis daarna – die wisselwerking tussen gekontroleerde en ongekontroleerde leer; anders gestel, leerprobleme waarvoor daar 'n klein hoeveelheid geëtiketteerde afrigtingsdata en 'n groot hoeveelheid ongeëtiketteerde afrigtings-data beskikbaar is (Ghahramani, 2004; Zhu & Goldberg, 2009). Gedeeltelik-gekontroleerde klassifikasie is 'n spesiale tipe klassifikasie, aangesien tradisionele klassifiseerders (afdeling 2.1.1) net van geëtiketteerde afrigtingsdata gebruik maak. Geëtiketteerde voorbeelde is dikwels moeilik en duur om te bekom aangesien dit die insette van ervare menslike annoteerders vereis, terwyl ongeëtiketteerde voorbeelde relatief maklik bekombaar is. Gedeeltelik-gekontroleerde leer poog om beter klassifiseerders te bou deur gebruik te maak van beide geëtiketteerde en ongeëtiketteerde afrigtingsdata (Zhu, 2009). Vervolgens word 'n paar bekende gedeeltelik-gekontroleerde leeralgoritmes bespreek.

2.1.3.1 SELFLEER

Een van die vroegste voorbeelde wat van ongeëtiketteerde data vir klassifikasieprobleme gebruik maak, is *selfleer* (Eng. *self-training*) (Chapelle *et al.*, 2006). Selfleer is ’n iteratiewe proses waar die leerproses sy eie voorspellings gebruik om homself af te rig (Chapelle *et al.*, 2006; Zhu & Goldberg, 2009). Selfleer word in sommige gevalle ook na verwys as ’n skoenlusproses (Eng. *bootstrapping*) (Zhu & Goldberg, 2009). Die klassifiseerder word in die eerste stap met ’n klein aantal geëtiketteerde afrigtingsdata afgerig, waarna die klassifiseerder gebruik word om nuwe data te klassifiseer. Die klassifikasiegevalle met die nuutvoorspelde klassifikasies wat die hoogste vertrouenspunte het, word vervolgens by die afrigtingsdata gevoeg waarna die proses herhaal word (Zhu, 2009).

Die voordeel van selfleer is dat dit eenvoudig is. Die selfleerprosedure omvou die afrigter sonder om die binnewerk van die betrokke algoritme te beïnvloed. Die afrigter kan ’n eenvoudige algoritme wees of ’n komplekse klassifiseerder wat gebaseer is op ’n vooraf-vervaardigde toepassing waarop daar nie veranderings gemaak kan word nie.

Een van die nadele van selfleer is dat die moontlikheid bestaan dat ’n fout aan die begin van die prosedure versterk word deur met elke iterasie verkeerde klassifikasies te maak. Daar bestaan verskeie metodes wat poog om hierdie probleem te oorkom, byvoorbeeld om sekere ongeëtiketteerde punte te “ontleer” wanneer die vertrouenspunt onder ’n sekere drempel daal (Zhu, 2009; Zhu & Goldberg, 2009).

Voorbeelde van toepassings van selfleer sluit in woordbetekenisvereenduidiging (Eng. *word sense disambiguation*) (Yarowsky, 1995), sinsontleding (Eng. *parsing*) (Charniak, 1997) en spraakherkenning (Bacchiani *et al.*, 2006). Clark *et al.* (2003) pas selfleer op woordsoort-etikettering toe en rapporteer geen verbetering nie of slegs ’n geringe verbetering of in sekere gevalle selfs ’n verswakking in akkuraatheid.

2.1.3.2 DEELLEER

Deelleer (Eng. *co-training*) wat aanvanklik voorgestel is deur Blum en Mitchell (1998), maak gebruik van meerdere afrigters wat elk ’n ander aansig van die afrigtingsdata het. Wanneer een afrigter ’n hoë vertrouenspeil het oor ’n klassifikasie word die klassifikasiegeval met die voorspelde klassifikasie by die afrigtingsdata van die ander afrigters gevoeg. ’n Variant soos voorgestel deur Dasgupta *et al.* (2002) is om alleenlik die klassifikasiegeval by die afrigtingsdata te voeg in die gevalle waar meer as een afrigter saamstem oor ’n klassifikasie. So is daar ’n hoër kans dat die klassifikasie wat by die afrigtingsdata gevoeg word inderdaad korrek is. Nog ’n variant is om beperkings op die afrigtingsdata te plaas, soos om die klassifikasiegevalle in

pare voor te stel waar die afrigter nie weet wat die klassifikasie is nie, maar wel weet dat dit dieselfde is vir beide lede van ’n paar (Abney, 2007).

In die deelleermetode maak dit nie saak wat die leeralgoritmes van die onderskeie afrigters is nie, solank die klassifiseerders ’n vertrouenspunt aan die voorspelde klassifikasies kan toeken. Dit maak deelleer toepaslik vir menige taak. Die sukses van deelleer hang daarvan af of die onderskeie aansigte van die afrigtingsdata op sigself goeie klassifiseerders kan produseer. Verder moet die onderskeie aansigte voorwaardelik onafhanklik wees, gegewe die klassifikasie. Dit behels dat as klassifikasie y bekend is, dan sal aansig $x^{(1)}$ nie bepaal wat in aansig $x^{(2)}$ waargeneem word nie (Zhu & Goldberg, 2009).

Sarkar (2001) en Steedman *et al.* (2003) gebruik deelleer vir sinsontleding en lei uit hulle navorsing af dat deelleer effektiewer is wanneer klein hoeveelhede geëtiketteerde afrigtingsdata beskikbaar is. Deelleer word ook toegepas op tekseenheidonttrekking (Eng. *named entity extraction*) deur Jones (2005) wat vind dat deelleer nie goeie resultate lewer vir hulle spesifieke voorstelling nie.

2.1.3.3 GENERATIEWE MODELLE

Een van die oudste gedeeltelik-gekontroleerde leermetodes is generatiewe modelle (Eng. *generative models*). Generatiewe modelle neem aan daar bestaan ’n model $p(x,y) = p(y)p(x|y)$ waar $p(x|y)$ ’n identifiseerbare mengselverdeling is, soos byvoorbeeld Gauss-mengselmodelle (GMM) (Eng. *Gaussian Mixture Models*). Met behulp van groot hoeveelhede ongeëtiketteerde data word die mengselkomponente geïdentifiseer waarna daar in ideale gevalle slegs een geëtiketteerde voorbeeld per komponent nodig is om die mengselverdeling ten volle te bepaal. Die mengselkomponente kan geïdentifiseer word met groot hoeveelhede ongeëtiketteerde data (Zhu, 2009). Tydens afrigting is die doel om ’n goeie θ te vind, waar θ gebruik word om die versameling van alle parameters in $p(x|y)$ en $p(y)$ aan te dui, aldus $p(x|y,\ \theta)$ en $p(y|\theta)$. Dit kan bereik word deur die maksimumaanneemlikheidsberaming-parameter (Eng. *maximum likelihood estimate (MLE)*) waaronder die data-aanneemlikheid $p(\mathcal{D}|\theta)$ die grootste is te bereken, waar $\mathcal{D}$ die afrigtingsdata is. Daar bestaan menige optimeringsmetode wat poog om ’n lokale optimale θ te vind waarvan die belangrikste die iteratiewe prosedure genaamd die EM-algoritme (E staan vir Verwagte waarde en M staan vir Maksimering) is wanneer ongeëtiketteerde data ter sprake is (Zhu & Goldberg, 2009).

Voorbeelde van generatiewe modelle is die Gauss-mengselmodel, wat ’n algemene keuse vir kontinue eienskapvektore is en ook aan die belangrike vereiste van identifiseerbaarheid (Zhu & Goldberg, 2009:31) voldoen, asook die multinomialemengselmodel (Eng. *Multinomial Mixture Model*), wat ’n algemene keuse is waar diskrete vektore ter sprake is, soos vir teksklassifikasie.

Nog ’n voorbeeld van generatiewe modelle is versteekte Markov-modelle (Eng. *Hidden Markov Models (HMM)*).

Ongeëtiketteerde data kan ’n positiewe invloed op die resultate van die klassifiseerder tot gevolg hê, mits die mengselmodel se aanname korrek is. Indien die model onrealisties is, kan ongeëtiketteerde data ’n klassifiseerder slegter laat vaar as wanneer dit net met geëtiketteerde data afgerig is (Cozman *et al.*, 2003). Dit is daarom belangrik om die mengselmodel so saam te stel dat dit die realiteit reflekteer (Zhu, 2009). In die praktyk word mengselkomponente deur die EM-algoritme bereken. Die EM-algoritme is geneig om die lokale maksimum te verskaf en indien dit grootliks verskil van die globale maksimum, kan ongeëtiketteerde data weereens ’n negatiewe invloed op die resultate tot gevolg hê (Zhu, 2009).

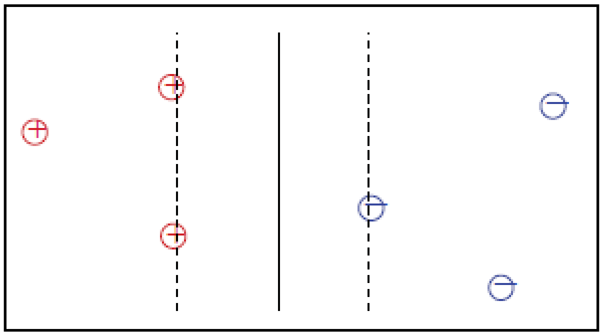
Nigam *et al.* (2000) pas die EM-algoritme op multinomialemengselmodelle vir die taak van teks-klassifikasie toe en toon dat die klassifiseerders beter vaar as dié wat net van geëtiketteerde afrigtingsdata gebruik gemaak het. Fujino *et al.* (2005) brei uit op generatiewe mengselmodelle deur gebruik te maak van die maksimum entropiebeginsel en ’n sydigheidskorreksieterm (Eng. *bias correction*), asook diskriminerende afrigting (Eng. *discriminative training*) by te voeg.

2.1.3.4 GEDEELTELIK-GEKONTROLEERDE STEUNVEKTORKLASSIFISEERDERS

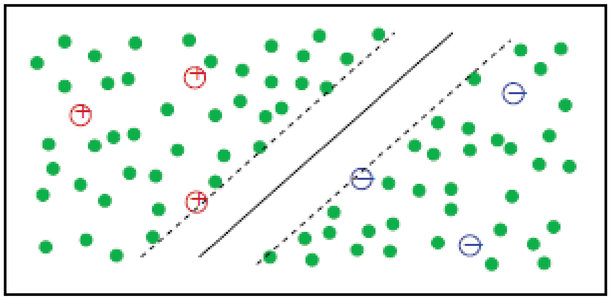
Gedeeltelik-gekontroleerde *Steunvektorklassifiseerders* (S3VM) (Eng. *Semi-Supervised Support Vector Machines*) is aanvanklik deur Vapnik (1998) voorgestel en deel die afrigtingsdata in gebiede op en poog dan om ’n *beslissingsgrens* (Eng. *decision boundary*) te vind wat verskillende klasse skei. Daar word aangeneem dat die beslissingsgrens nie deur gebiede van digte ongeëtiketteerde data sal gaan nie. Figuur 2.2 en Figuur 2.3 illustreer laasgenoemde stelling. Figuur 2.2 stel ’n volledig geëtiketteerde datastel voor met ’n reguit lyn in die middel wat twee klasse skei sodat die minimum van die afstande na die naaste positiewe en naaste negatiewe voorbeelde gemaksimeer is. Hierdie lyn is die liniêre beslissingsgrens wat deur steunvektorklassifiseerders gevind word. Die figuur wys ook twee stippellyne wat deur die naaste positiewe en negatiewe voorbeelde sny. Die afstand vanaf die beslissingsgrens tot by die stippellyne word die geometriese rand genoem en word gemaksimeer deur steunvektor-klassifiseerders. In die geval waar daar geëtiketteerde, sowel as ongeëtiketteerde data voorkom, kan die beslissingsgrens wat gebaseer is op slegs geëtiketteerde data, soos voorgestel in Figuur 2.2, moontlik deur die digte ongeëtiketteerde gebiede, soos voorgestel in Figuur 2.3, deursny. Die nuwe beslissingsgrens in Figuur 2.3, wat nou gedefinieer is deur beide geëtiketteerde en ongeëtiketteerde data, skei steeds die twee klasse in die geëtiketteerde data,

maar ook die twee klasse in die ongeëtiketteerde data. Let ook op dat die geometriese rand nou kleiner is as in Figuur 2.2

.



Figuur 2.2 – Steunvektorklassifiseerdersbeslissingsgrens. Geneem uit Zhu en Goldberg (2009)



Figuur 2.3 – S3VM beslissingsgrens. Geneem uit Zhu en Goldberg (2009)

Die beslissingsgrens deursny die eienskapruimte in twee helftes genaamd die positiewe en negatiewe helftes. Die betekende afstand (Eng. *signed distance*) van ’n geëtiketteerde voorbeeld is positief as ’n voorbeeld in die positiewe kant val en negatief as ’n voorbeeld in die negatiewe kant van die beslissingsgrens val. Die betekende geometriese rand is die afstand vanaf die beslissingsgrens tot by die naaste geëtiketteerde voorbeeld. Wanneer ’n beslissingsgrens die geëtiketteerde afrigtingsvoorbeelde volkome skei (met ander woorde die afrigtingsdata is lineêr skeibaar), sal die geometriese rand positief wees. Die mikpunt is om ’n beslissingsgrens te vind wat die geometriese rand sal maksimeer. Wanneer die afrigtingsdata lineêr skeibaar (Eng. *linearly separable*) is, sal daar ten minste een liniêre beslissingsgrens wees wat alle geëtiketteerde voorbeelde kan skei sodat hulle aan die regte kant van die beslissingsgrens val.

Vir die geval waar die afrigtingsdata nie lineêr skeibaar is nie, word daar spelingsveranderlikes (Eng. *slack variables*) voorgestel wat die hoeveelheid toegelate verslapping vir elke voorbeeld verteenwoordig. In hierdie geval word sommige afrigtingsvoorbeelde toegelaat om aan die verkeerde kant van die beslissingsgrens voor te kom, terwyl daar steeds gepoog word om ’n

maksimum randskeiding te vind. In die geval van geëtiketteerde afrigtingsdata is dit moontlik om te bepaal of ’n voorbeeld aan die regte of verkeerde kant van die beslissingsgrens val deur middel van die skarnierverliesfunksie (Eng. *hinge loss function*) wat voorbeelde wat aan die regte kant van die beslissingsgrens val, maar binne die rand, penaliseer en voorbeelde wat aan die verkeerde kant van die beslissingsgrens val selfs nog meer penaliseer. In die geval waar daar ongeëtiketteerde afrigtingsdata teenwoordig is waarvoor dit nie moontlik is om te bepaal of ’n voorbeeld aan die positiewe of negatiewe kant van die beslissingsgrens val nie, word die beraamde annotasie, of te wel die *veronderstelde annotasie* (Eng. *putative label*) van die voorbeeld gebruik om die hoedverliesfunksie (Eng. *hat loss function*) toe te pas. Die hoedverliesfunksie is op die skarnierverliesfunksie gebaseer, maar aangesien die veronderstelde annotasies gebruik word, sal die ongeëtiketteerde voorbeelde altyd aan die regte kant van die beslissingsgrens val. Die hoedverliesfunksie penaliseer ongeëtiketteerde voorbeelde wat binne die rand val en waarvan die voorspeller dus onseker is, terwyl dit voorkeur gee aan voorbeelde wat buite die rand, vêr weg van die beslissingsgrens, val en waarvan die voorspeller dus sekerder is.

Terselfdertyd moet die beslissingsgrens binne ’n lae digtheidsgaping binne die datastel val, sodat so min as moontlik ongeëtiketteerde voorbeelde naby aan die grens is, met onseker voorspellers. In die praktyk kom daar egter soms gevalle voor waar die meerderheid, of selfs soms alle, ongeëtiketteerde voorbeelde se voorspelling binne slegs een klas val. Om hierdie rede word die hoeveelheid ongeëtiketteerde voorbeelde wat aan ’n spesifieke klas toegeken mag word proporsioneel bepaal volgens die hoeveelheid voorbeelde wat aan daardie klas toegeken is in die geëtiketteerde data.

’n Uitdaging van S3VM teenoor gewone steunvektorklassifiseerders is dat S3VM nie-konveks is, terwyl steunvektorklassifiseerders konveks is. Dit is relatief maklik om ’n konvekse funksie te minimeer aangesien dit ’n goed gedefinieerde onderste grens het. Aan die ander kant is S3VM nie-konveks met meervoudige lokale minimums waar ’n leeralgoritme kan vasval in ’n sub-optimale lokale minimum en nooit die globale minimum oplossing vind nie.

Nog ’n uitdaging van S3VM is dat die aanname gemaak word dat die klasse goed geskei is sodat die beslissingsgrens binne ’n lae digtheidsruimte in die eienskapruimte val en nie deur digte ongeëtiketteerde data sny nie. Indien hierdie aanname nie geld nie, kan dit gebeur dat die algoritme op ’n dwaalspoor beland en nie goeie resultate lewer nie (Zhu & Goldberg, 2009).

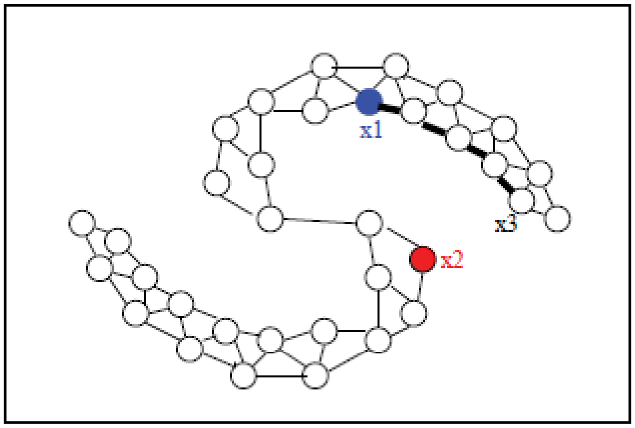
Een van die eerste wydgebruikte implementerings van S3VM wat verbeterde resultate lewer vir klein en groot datastelle is dié van Joachims (1999b) wat toegepas word op teksklassifikasie waar dokumente aan ’n spesifieke semantiese kategorie of kategorieë toegeken word.

2.1.3.5 GRAFIEKGEBASEERDE METODES

In Grafiekgebaseerde metodes (Eng. *Graph-Based Methods*) word data voorgestel deur die nodusse van ’n grafiek. Die grafiekgebaseerde gedeeltelik-gekontroleerde afrigting begin deur ’n grafiek uit die afrigtingsdata te konstrueer waar die nodusse die geëtiketteerde en ongeëtiketteerde klassifikasiegevalle voorstel. Sodra die grafiek saamgestel is, sal afrigting die toekenning van klassifikasies aan die nodusse behels. Dit word moontlik gemaak deur die rande wat geëtiketteerde en ongeëtiketteerde nodusse verbind en gewoonlik sonder rigting is. ’n Rand tussen twee nodusse stel die gelyksoortigheid van die twee klassifikasiegevalle voor met behulp van ’n gewig. As die gewig van die rand groot is, dan word verwag dat die klassifikasies van die twee voorbeelde dieselfde is. Die gewig kan deur verskeie metodes bereken word wat onder andere insluit die berekening van die Euklidiese afstand waar die gewig afneem soos die afstand toeneem (Zhu & Goldberg, 2009). Formeel gestel word funksie f beraam op ’n grafiek sodat f gelyktydig aan twee kriteria voldoen:

1) Die voorspelling $f(x)$ is naby aan die gegewe klassifikasie y op die geëtiketteerde nodusse;
2) f is glad (Eng. *smooth*) op die hele grafiek.

Dit word bereik in ’n reguleringsraamwerk waar 1) dui op ’n verliesfunksie en 2) dui op ’n reguleerder (Eng. *regulariser*) (Zhu, 2009; Zhu & Goldberg, 2009). Punt 2) beteken dat die klassifikasies met betrekking tot die grafiek “glad” sal wees, sodat dit min sal varieer op die grafiek. Wanneer twee klassifikasiegevalle derhalwe deur ’n sterk rand verbind word, neig hulle klassifikasies om dieselfde te wees (Zhu & Goldberg, 2009). Figuur 2.4 vertoon ’n gekonstrueerde grafiek met geëtiketteerde klassifikasiegevalle $x_1 \text{ en } x_2$ en ongeëtiketteerde klassifikasiegeval x_3. Die ongeëtiketteerde klassifikasiegeval x_3 sal meer deur die klassifikasie van x_1 beïnvloed word as deur die klassifikasie van x_2 aangesien x_1 nader aan x_3 in die grafiek is, al is x_2 nader in Euklidiese afstand.



Figuur 2.4 – ’n Grafiek wat gekonstrueer is uit geëtiketteerde en ongeëtiketteerde klassifikasiegevalle. Geneem uit Zhu en Goldberg (2009)

’n Aantal grafiekgebaseerde gedeeltelik-gekontroleerde leeralgoritmes, waarvan die belangrikste verskille die keuse van verliesfunksie en regulariseerder behels, kom in die literatuur voor. Van die belangrikste leeralgoritmes sluit onder andere in Mincut, Harmoniese funksie (Eng. *Harmonic function*) en veelvuldige regulering (Eng. *manifold regularisation*) (Zhu, 2009; Zhu & Goldberg, 2009).

Grafiekgebaseerde gedeeltelik-gekontroleerde leer is reeds effektief in die praktyk vir onder andere teksklassifikasie gebruik en is relatief maklik paralleliseerbaar en skaalbaar vir groot hoeveelhede data. Dit bied ’n uniforme voorstelling vir heterogene data terwyl menige datastelle natuurlik deur ’n grafiek voorgestel kan word (Subramanya & Talukdar, 2012). Die grafiek-gebaseerde gedeeltelik-gekontroleerde leer kan dus goeie resultate lewer mits die grafiek geskik is vir die betrokke taak. Wanneer die grafiek swak gekonstrueer is sal die resultate swak wees aangesien die metode sensitief is met betrekking tot die grafiekstruktuur en randgewigte (Blitzer & Zhu, 2008).

Voorbeelde van grafiekgebaseerde gedeeltelik-gekontroleerde leer sluit onder meer in Blum en Chawla (2001) wat die Mincut-metode voorstel en toepas op klassifikasieprobleme in verskillende datastelle. Pang en Lee (2004) gebruik ook Mincut vir die klassifikasie van “subjektiwiteit” deur ’n model af te rig wat onderskei tussen rolprentopsommings en rolprent-resensies. Nadat slegs die subjektiewe sinne in ’n dokument geïdentifiseer is deur middel van Mincut, word hierdie sinne gebruik om die dokumentsentiment as positief of negatief te klassifiseer. Niu *et al.* (2005) gebruik die Harmoniese funksie vir woordbetekenis-vereenduidiging, terwyl ’n vername voorbeeld van die toepassing van veelvuldige regulering die navorsing van Belkin *et al.* (2006) op teksklassifikasie is. Grafiekgebaseerde gedeeltelik-gekontroleerde leer word ook met sukses op die probleem van woordsoortetikettering toegepas (Subramanya *et al.*, 2010).

Vervolgens word die tegniek van die kombinering van verskeie algoritmes vir die klassifikasietaak bespreek.

2.2 KOMBINASIE

’n *Kombinasieklassifiseerder* is die kombinering van verskillende *eerstevlakklassifiseerders* met die oog daarop om ’n beter klassifikasie in ’n spesifieke klassifikasiegeval te maak. ’n *Kandidaat* is ’n klassifikasie wat deur ’n eerstevlakklassifiseerder binne ’n kombinasieklassifiseerder voorgestel word, maar wat nie noodwendig deur die hele kombinasiesisteem as die klassifikasie en uiteindelike uitvoer vir die klassifikasiegeval aangewys word nie. ’n Kandidaat kan ook ’n klassifikasie wees wat deur ’n eerstevlakklassifiseerder as klassifikasie oorweeg word, maar ook nie noodwendig as uiteindelike klassifikasie aangewys word nie.

Gegee Afrikaanse woordsoortetikettering (Pilon, 2005) as klassifikasietaak, dan is die woordsoortetiketteerder die klassifiseerder. Gestel die woordsoortetiketteerder moet die woord “kleurvolle” in die sin “Die dogtertjie pluk die kleurvolle blom.” etiketteer. As invoer ontvang die woordsoortetiketteerder ’n eienskapvektor. Hierdie eienskapvektor kan uit net ’n enkele woord bestaan, byvoorbeeld $\{Tekseenheid = \text{kleurvolle}\}$, en dus atomies van aard wees of dit kan meer inligting bevat soos in (2.9), in welke geval dit ’n komplekse eienskapvektor sal wees. Die woordsoortetiketteerder sal nou uit ’n eindige lys woordsoortetikette ’n woordsoortetiket vir die spesifieke klassifikasiegeval kies en gevolglik word die klassifikasie “ASA” aan “kleurvolle” toegeken.

$$\{Tekseenheid = kleurvolle, Vorigetekseenheid = die, Volgendetekseenheid = blom\} \tag{2.9}$$

Gestel die woordsoortetiketteerder is ’n kombinasieklassifiseerder wat ’n aantal woordsoortetiketteerders kombineer. Elke individuele woordsoortetiketteerder sal dan die klassifikasiegeval klassifiseer, dus etiketteer. Daarna stel elke individuele woordsoortetiketteerder ’n woordsoortetiket, genoem ’n kandidaat, aan die kombinasie-woordsoortetiketteerder voor. Die kombinasie-woordsoortetiketteerder neem dan alle kandidate in ag en lewer uiteindelik ’n enkele woordsoortetiket of klassifikasie as uitvoer.

Die eerstevlakklassifiseerders waaruit ’n kombinasieklassifiseerder bestaan kan op meer as een manier van mekaar verskil. Verskillende leeralgoritmes kan tydens die afrigtingsfase gebruik word, of dieselfde leeralgoritme wat telkens verskillende eienskappe gebruik, kan gebruik word, of die verskillende eerstevlakklassifiseerders kan met verskillende afrigtingsdata afgerig word. Hoe dit ook al sy – die belangrike is dát die eerstevlakklassifiseerders van mekaar verskil,

eerder as hóé hulle van mekaar verskil. Die onderliggende idee van ’n kombinasie-klassifiseerder is dat, wanneer die foute van die eerstevlakklassifiseerders ongekorreleerd genoeg is en die persentasie foute wat hulle maak laag genoeg is, die kombinasie-klassifiseerder beter sal vaar as wat enige van die eerstevlakklassifiseerders waaruit dit saamgestel is in afsondering sal vaar (Brill & Wu, 1998; Dietterich, 1997; Hoste & Daelemans, 2000; Karahan *et al.*, 2003; Utsuro *et al.*, 2002; Van Halteren *et al.*, 2001).

Kombinasieklassifiseerders word in verskillende natuurliketaalverwerkingtake gebruik. Van hierdie take behels onder andere woordsoortetikettering (Van Halteren *et al.*, 2001), klassifikasie in die herkenning van gesproke taal (Karahan *et al.*, 2003), naamwoordstuk-herkenning (Sang *et al.*, 2000), tekseenheidonttrekking (Utsuro *et al.*, 2002), grafeem na foneemomskakeling (Van Halteren, 1999), sinsontleding (Henderson & Brill, 1998), masjien-vertaling (Frederking & Nirenburg, 1994), ens.

Van die bekendste kombinasiemetodes word volgende bepreek.

2.2.1 KASKADEMETODE

Karahan *et al.* (2003) maak gebruik van die kaskadekombinasiemetode (Eng. *cascading*) vir klassifikasie in die herkenning van gesproke taal (Eng. *call classification*). In hierdie metode word klassifiseerders gekombineer deur die klassifiseerders agtermekaar op te lyn. Tydens klassifikasie kry die klassifiseerder wat eerste opgelyn is die geleentheid om te klassifiseer. Indien die klassifiseerder nie ’n vertrouenspunt wat hoër is as ’n vasgestelde limietwaarde aan ’n kandidaat kan toeken nie, beweeg die sisteem aan na die volgende klassifiseerder in die ry. As die volgende klassifiseerder dieselfde kandidaat kies as die vorige een, word die kandidaat se vertrouenspunt opgestoot na ’n punt wat hoër is as die vasgestelde limietwaarde. Omdat die volgorde waarin klassifiseerders opgelyn word die akkuraatheid van die uiteindelike kombinasie-klassifiseerder beïnvloed, is dit dus beter om die klassifiseerders volgens hulle individuele akkuraatheidsyfers op te lyn waar dié met die hoogste akkuraatheidsyfer eerste in lyn geplaas word.

2.2.2 STEMMETODE

Waarskynlik die eenvoudigste metode waarvolgens klassifiseerders in natuurliketaalverwerking gekombineer kan word, is die stemmetode. Oor die algemeen word twee tipes stemmetode gebruik, naamlik eenvoudige stemming en geweegde stemming.

2.2.2.1 EENVOUDIGE STEMMING

Volgens hierdie metode kry elke klassifiseerder ’n gelyke stem. Lin *et al.* (2003) onderskei tussen twee tipes eenvoudige stemming genaamd meerderheidstemming (Eng. *majority voting*) en pluraliteitstemming (Eng. *plurality voting*). Die twee tipes verskil van mekaar in die manier waarop ’n klassifikasie gekies word. Beide ken aan elke eerstevlakklassifiseerder ’n gelyke stem toe. Vir ’n kandidaat om as klassifikasie gekies te word deur middel van meerderheidstemming moet daardie kandidaat meer as die helfte van alle eerstevlakklassifiseerders se stemme werf. In die geval van pluraliteitstemming word die kandidaat met die meeste stemme as klassifikasie gekies, ongeag hóéveel meer stemme daardie kandidaat as die ander kandidate gehad het. Na verdere studie uit die literatuur, blyk dit dat baie navorsers nie onderskei tussen meerderheid-stemming en pluraliteitstemming nie. Die term meerderheidstemming word meer dikwels gebruik, alhoewel daar in werklikheid gereeld eerder van pluraliteitstemming gebruik gemaak word. In die geval waar kandidate ’n gelyke hoeveelheid stemme het, is daar verskillende maniere om die gelykopstryd op te los. Van Halteren *et al.* (2001) byvoorbeeld, kies ewekansig een van die kandidate wat gelyke stemme ontvang het.

2.2.2.2 GEWEEGDE STEMMING

In geweegde stemming (Eng. *weighted voting*) word elke eerstevlakklassifiseerder se stem vermenigvuldig met ’n gewig wat aan daardie klassifiseerder toegeken is. Wanneer ’n eerstevlakklassifiseerder dus vir ’n kandidaat stem, ontvang daardie kandidaat die klassifiseerder se stemgewig as ’n vertrouenspunt. Wanneer meer as een klassifiseerder vir dieselfde kandidaat stem, word hulle stemgewigte bymekaargetel en toegeken aan die kandidaat as ’n vertrouenspunt. Uiteindelik word die kandidaat met die hoogste vertrouenspunt as die klassifikasie gekies (Karahan *et al.*, 2003; Van Halteren *et al.*, 2001).

Die gewig wat aan elke eerstevlakklassifiseerder toegeken word, hou gewoonlik verband met daardie klassifiseerder se akkuraatheid. Van Halteren *et al.* (2001) onderskei tussen drie geweegde stemmingmetodes:

In die eerste metode ontvang elke eerstevlakklassifiseerder sy totale presisie (Eng. *total precision*) as stemgewig. Wanneer ’n klassifiseerder dan moet stem vir ’n kandidaat, kry die kandidaat waarvoor die klassifiseerder stem die totale presisie van die klassifiseerder as vertrouenspunt. Elke eerstevlakklassifiseerder se gewig word dus bereken volgens formule (2.10).

$$Gewig = 1 \times totale\ presisie \qquad (2.10)$$

In die tweede metode ontvang elke eerstevlakklassifiseerder die kandidaatpresisie van die gekose kandidaat by daardie klassifiseerder as gewig (Van Halteren *et al.*, 2001). Vir elke klassifiseerder bestaan daar vir elke moontlike kandidaat ’n kandidaatpresisie wat vooraf bereken is deur middel van toetsdata. ’n Kandidaat se kandidaatpresisie by ’n klassifiseerder toon aan hoe akkuraat daardie klassifiseerder daardie kandidaat geklassifiseer het op die toets-data. Elke eerstevlakklassifiseerder se gewig word dus bereken volgens die volgende formule (2.11)

$$Gewig = 1 \times kandidaatpresisie \tag{2.11}$$

byvoorbeeld, gegee ’n kombinasie-woordsoortetiketteerder wat uit twee woordsoort-etiketteerders bestaan, Klassifiseerder A en Klassifiseerder B. Die kombinasie-woordsoort-etiketteerder ontvang ’n invoerwoord “’n” wat geklassifiseer moet word. Klassifiseerder A kies die woordsoortetiket *bepaalde lidwoord* as kandidaat, terwyl Klassifiseerder B *onbepaalde lidwoord* as kandidaat kies. Nou moet die klassifiseerders stem vir die korrekte kandidaat. Gestel Klassifiseerder A het ’n kandidaatpresisie van 0.7 op bepaalde lidwoorde en Klassifiseerder B het ’n kandidaatpresisie van 0.9 op onbepaalde lidwoorde. Klassifiseerder A kry dan ’n stemgewig van 0.7 en Klassifiseerder B ’n stemgewig van 0.9. Dit het tot gevolg dat die kandidaat, *bepaalde lidwoord,* ’n vertrouenspunt van 0.7 ontvang, terwyl die kandidaat, *onbepaalde lidwoord,* ’n vertrouenspunt van 0.9 ontvang. Klassifiseerder B se kandidaat het dus ’n hoër vertrouenspunt en die invoerwoord word geëtiketteer as ’n *onbepaalde lidwoord*.

In die derde metode word elke klassifiseerder se gewig bereken deur van presisie en herroeping (afdeling 1.6.3) gebruik te maak. Elke klassifiseerder se gewig word bereken volgens die volgende formule (2.12).

$$Gewig = kandidaatpresisie + (1 - kandidaatherroeping) \tag{2.12}$$

Ter verduideliking: Gegee twee klassifiseerders, Klassifiseerder A en Klassifiseerder B. Vir ’n spesifieke klassifikasiegeval stel Klassifiseerder A $kandidaat_x$ voor terwyl Klassifiseerder B vir dieselfde klassifikasiegeval $kandidaat_y$ voorstel. Klassifiseerder A se gewig word bereken deur die kandidaatpresisie van Klassifiseerder A op $kandidaat_x$ by 1 minus die kandidaatherroeping van Klassifiseerder B op $kandidaat_x$ te tel. ’n Kandidaat se kandidaatherroeping by ’n klassifiseerder dui aan hoe gereeld die klassifiseerder op die toetsdata die spesifieke kandidaat korrek as klassifikasie gekies het. ’n Kandidaatherroeping van 1 sal beteken dat die klassifiseerder elke keer die korrekte kandidaat as klassifikasie aangewys het, terwyl ’n

kandidaatherroeping van 0 sal beteken dat die klassifiseerder nie 1 keer die korrekte kandidaat as klassifikasie aangewys het nie. Die laasgenoemde formule word fyner voorgestel in (2.13).

$$GewigA = kandidaatpresisieA + (1 - kandidaatherroepingB) \tag{2.13}$$

Hoe meer gereeld Klassifiseerder B (wat in hierdie geval $kandidaat_y$ voorstel) klassifikasie-gevalle wat werklik $kandidaat_x$ is korrek klassifiseer, hoe minder vertroue word in Klassifiseerder A gestel wanneer dit $kandidaat_x$ voorstel.

2.2.3 Versterking

Versterking (Eng. *Boosting*) is ’n metode wat enige gegewe masjienleeralgoritme se akkuraatheid kan verbeter (Schapire, 2003b). Die metode gebruik dieselfde algoritme om verskillende klassifiseerders af te rig. Die idee is om ’n groot aantal onakkurate klassifiseerders te kombineer om uiteindelik een akkurate klassifiseerder te maak (Karahan *et al.*, 2003; Marquez, 2000). Hoste en Daelemans (2000) verduidelik die proses van versterking soos volg: Die eerste klassifiseerder word afgerig deur die hele stel afrigtingsdata te gebruik en alle gewigte gelyk te stel. Die klassifiseerder word dan met behulp van toetsdata getoets en foute word uitgewys. Daarna word ’n volgende klassifiseerder afgerig, maar die verskil is dat die voorheen verkeerd geklassifiseerde voorbeelde nou groter gewig dra as die voorbeelde wat deur die vorige klassifiseerder korrek geklassifiseer is. Op hierdie manier word die nuwe klassifiseerder tydens die afrigtingsfase gedwing om meer aandag aan die “moeilike” voorbeelde te gee. Hierdie proses van versterking gaan voort totdat ’n vooraf bepaalde aantal iterasies bereik is. Die resultaat is ’n versterkte klassifiseerder wat saamgestel is uit ’n aantal onakkurate klassifiseerders. Freund en Schapire (1996) het in hulle navorsing die AdaBoost-algoritme ontwikkel wat op versterking gebaseer is. Trae versterking (Eng. *lazy boosting*) is ’n vereenvoudigde aanpassing op die AdaBoost-algoritme wat die lys eienskappe wat as invoer gegee word wanneer elke onakkurate klassifiseerder afgerig word, verminder. Hierdie aanpassing verbeter die effektiwiteit van die algoritme, terwyl die akkuraatheid van die versterkte klassifiseerder nie afneem nie (Escudero *et al.*, 2000).

2.2.4 Groepering

Breiman (1996) stel groepering (Eng. *bagging*) voor as ’n metode waarin meerdere weergawes van ’n klassifiseerder gegenereer word en dan saamgevoeg word om ’n saamgevoegde

klassifiseerder te vorm. Hierdie saamgevoegde klassifiseerder kan ook as ’n kombinasie-klassifiseerder gesien word, aangesien dit uit ’n aantal eerstevlakklassifiseerders saamgestel word. Net soos in die geval van versterking, gebruik groepering ook dieselfde algoritme om die eerstevlakklassifiseerders af te rig. Die klassifiseerders verskil van mekaar in dié opsig dat verskillende substelle van die afrigtingsdata gebruik word om die verskillende eerstevlak-klassifiseerders af te rig.

Groepering maak gebruik van skoenlussteekproefneming (Eng. *bootstrapping*) (Efron & Tibshirani, 1994) wat ewekansig gekose seleksies van die afrigtingsdata genereer. In Breiman (1996) is die grootte van elke seleksie gelyk aan die grootte van die oorspronklike stel afrigtingsdata. Só ’n seleksie kan ook kleiner of selfs groter as die oorspronklike stel afrigtingsdata wees. Gestel die oorspronklike stel afrigtingsdata bestaan uit N voorbeelde. Skoenlussteekproefneming sal dan ewekansig uit die oorspronklike stel N voorbeelde met *vervanging* trek. Dit beteken as voorbeeld x_n getrek word om as voorbeeld in die nuwe seleksie afrigtingsdata te dien, dan word x_n weer teruggeplaas in die oorspronklike stel, sodat daar in die volgende rondte van seleksie weer ’n kans bestaan dat dieselfde x_n getrek kan word. Op hierdie manier word daar ’n nuwe stel afrigtingsdata genereer waarin die moontlikheid bestaan dat ’n voorbeeld wat net een keer in die oorspronklike stel afrigtingsdata voorkom, meer as een keer, of glad nie in die nuwe stel voorkom nie.

Die uiteindelike eerstevlakklassifiseerders wat deur middel van die verskillende substelle ontstaan, word dan gekombineer deur van meerderheidstemming (Hoste & Daelemans, 2000) of van pluraliteitstemming (Breiman, 1996) gebruik te maak.

Net soos in die geval van versterking is die idee van groepering om ’n groot aantal onakkurate klassifiseerders te kombineer. Alhoewel Breiman (1996) 50 klassifiseerders in sy eksperimente gebruik, bevind hy dat die meeste verbetering by 10 klassifiseerders gewaar word en dat meer as 25 klassifiseerders oorbodig is. Hoste en Daelemans (2000) gebruik weer in hulle eksperimente net 10 klassifiseerders.

In groepering word gepoog om ’n stel eerstevlakklassifiseerders te skep wat in lyn is met die oorspronklike afrigtingsdata, maar wat tog van mekaar verskil. Die verdeling van die oorspronklike afrigtingsdata moet ooreenstem met die verdeling van die totale aantal skoenlus-steekproewe saam (Henderson & Brill, 2000).

Hoste en Daelemans (2000) vergelyk verder in hulle eksperimente versterking met groepering en kom tot die gevolgtrekking dat versterking beter vaar met nie-tipiese gevalle (Eng. *low-typical cases*), terwyl groepering weer beter vaar met meer tipiese gevalle (Eng. *typical cases*). Dit beteken dat ’n kombinasieklassifiseerder wat versterking gebruik beter sal vaar op data wat baie uitsonderings, of nie-tipiese gevalle bevat, terwyl ’n kombinasieklassifiseerder wat groepering

gebruik beter sal vaar op data wat meer tipiese gevalle bevat – dit is “normale” data, sonder baie uitsonderlike gevalle.

2.2.5 OPSTAPELING

’n Meer gesofistikeerde metode as die stemmetode is die metode van opstapeling (Eng. *stacking*). Dieselfde eerstevlakklassifiseerders as wat by die stemmetode gebruik word, kan ook by opstapeling gebruik word. Daar is egter ’n verskil in die manier waarop ’n klassifikasie gekies word. In plaas daarvan om vir die regte kandidaat te stem, word elke klassifiseerder se voorstel nou na ’n tweedevlakklassifiseerder gestuur (Karahan *et al.*, 2003; Van Halteren *et al.*, 2001). Hierdie tweedevlakklassifiseerder word vooraf afgerig om die regte klassifikasie te voorspel. Die invoer van die tweedevlakklassifiseerder kan beperk word tot die uitvoer van die eerstevlak-klassifiseerders, dus elke eerstevlakklassifiseerder se voorstel van ’n kandidaat, of dit kan nog addisionele inligting vanaf die oorspronklike invoerpatroon bevat. Addisionele inligting kan byvoorbeeld die invoerwoord, ’n vorige of ’n opeenvolgende invoerwoord, of ’n vorige of ’n opeenvolgende klassifikasie wees (Van Halteren *et al.*, 2001). Die tweedevlakklassifiseerder kies sy uitvoer op die basis van patrone wat voorkom by die uitvoer van die verskillende eerstevlakklassifiseerders en die addisionele inligting. Op hierdie manier word nie alleenlik die individuele invloed van elke eerstevlakklassifiseerder verminder nie, maar word die keuse van uitvoer deur die eerstevlakklassifiseerder in die identifikasie van die régte uitvoer deur die tweedevlakklassifiseerder benut. Hierdie metode maak dit selfs moontlik om kollektiewe foute reg te stel (Van Halteren *et al.*, 2001). Waar dit by die stemmetode nooit moontlik was vir ’n kombinasieklassifiseerder om ’n ander kandidaat te kies as wat deur een van die eerstevlak-klassifiseerders voorgestel is nie, is dit nou moontlik vir ’n tweedevlakklassifiseerder om ’n ander kandidaat te kies as wat deur enige van die eerstevlakklassifiseerders voorgestel is. Die keuse van ’n tweedevlakleerderalgoritme is, soos in die geval van ’n eerstevlakleerderalgoritme, baie wyd. Karahan *et al.* (2003) maak onder andere gebruik van liniêre regressie, logistiese regressie en besluitnemingsbome. Van Halteren *et al.* (2001) gebruik geheue-gebaseerde leer, maksimum entropie en ook besluitnemingsbome. Behalwe vir hierdie drie bekende masjienleer-metodes, gebruik hulle ook opgestapelde waarskynlikheidstemming.

Die voordeel van ’n opgestapelde sisteem, anders as by stemming, is dat daar meer inligting aan die tweedevlakklassifiseerder deurgegee word, byvoorbeeld in die geval van ’n woordsoort-etiketteerder sal die woord wat geëtiketteer moet word ook as invoer vir die tweedevlak-klassifiseerder gegee word, en nie net die verskillende eerstevlakklassifiseerders se voorstelle van kandidate nie.

Van Halteren *et al.* (2001) bevind dat alle opgestapelde sisteme beter vaar as sisteme wat gebruik maak van ’n stemmetode, mits daar genoeg afrigtingsdata bestaan. In die geval waar daar nie genoeg afrigtingsdata beskikbaar is nie, moet daar teruggeval word op minder gesofistikeerde metodes. In Van Halteren *et al.* (1998) word die metode van opgestapelde waarskynlikheidstemming (wat vervolgens bespreek word) gereken onder ’n stemmetode en dus as “minder gesofistikeerd” gereken. Hierdie metode vaar die beste op wat hulle reken ’n onvoldoende hoeveelheid data (116 000 woorde) is.

2.2.5.1 OPGESTAPELDE WAARSKYNLIKHEIDSTEMMING

Die opgestapelde waarskynlikheidstemming (Eng. *Stacked Probabilistic Voting*), of te wel die TagPair-metode, is reeds met sukses deur Van Halteren *et al.* (1998) toegepas. Gestel Klassifiseerder A stel $kandidaat_x$ voor en Klassifiseerder B stel $kandidaat_y$ voor: die TagPair-metode bereken dan die waarskynlikheid dat $kandidaat_z$ in hierdie situasie eintlik die wenkandidaat sal wees. Dit word voorgestel deur die volgende formule (2.14).

$$P(kandidaat_z|kandidaat_x, kandidaat_y) \tag{2.14}$$

Van Halteren *et al*. (1998) het hierdie metode as ’n stemmetode geklassifiseer, maar in Van Halteren *et al*. (2001) word dit, alhoewel daar gestem word, erken as ’n opgestapelde metode juis omdat ’n TagPair-klassifiseerder nie noodwendig een van die kandidate kies wat deur die eerstevlakklassifiseerders voorgestel is nie.

Klassifikasie vind plaas deurdat elke twee eerstevlakklassifiseerders (eerstevlakklassifiseerder-paar) toegelaat word om vir elke moontlike kandidaat te stem. Byvoorbeeld, in die geval van ’n woordsoortetiketteerder sou daar vir elke moontlike woordsoortetiket wat bestaan in ’n spesifieke woordsoortetiketteerder gestem word. Die stemgewig wat aan hierdie paar toegeken word, is gelyk aan $P(kandidaat_z|kandidaat_x, kandidaat_y)$. Dit wil sê gestel Klassifiseerder A stel $kandidaat_x$ voor en Klassifiseerder B stel $kandidaat_y$ voor, wat is die waarskynlikheid dat die regte kandidaat eintlik $kandidaat_z$ is? Hierdie waarskynlikheid P is die vertrouenspunt wat aan $kandidaat_z$ toegeken word. Indien die kombinasie van $kandidaat_x$ en $kandidaat_y$ nie voorgekom het in die afrigtingsdata nie, word daar teruggeval op die kennis wat tydens die afrigtingsfase oor die individuele eerstevlakklassifiseerder opgedoen is, en word ’n stemgewig wat gelyk is aan $P(kandidaat_z|kandidaat_x)$ aan die eerstevlakklassifiseerder toegeken. Dit wil sê gestel Klassifiseerder A stel $kandidaat_x$ voor, wat is die waarskynlikheid dat die regte

kandidaat eintlik $kandidaat_z$ is? Hierdie stemmetode maak dit moontlik vir 'n minderheids-kandidaat om aangewys te word as die wenkandidaat.

2.2.5.2 GEWEEGDE WAARSKYNLIKHEIDSVERDELINGSTEMMING

Anders as TagPair is geweegde waarskynlikheidsverdelingstemming (WPDV) (Eng. *Weighted Probability Distribution Voting*) nie beperk tot pare-eienskappe nie, maar kan dit die waarskynlikheidverdelings van alle eienskapkombinasies wat voorkom in die afrigtingsdata gebruik (Van Halteren *et al.*, 2001).

Gedurende die afrigtingsfase bepaal WPDV die klassifikasiewaarskynlikheidverspreiding van elke atomiese en komplekse eienskapvektor wat ingevoer word. Gedurende klassifikasie neem WPDV dan alle eienskapvektore wat in die nuwe invoer voorkom en ken die ooreenstemmende waarskynlikheidverspreiding, vermenigvuldig met 'n gewig wat verband hou met die eienskapvektor, toe. Die geweegde verspreidings van elke kandidaat word opgetel en die kandidaat met die hoogste somtotaal word aangewys as die klassifikasie. Die volgende is 'n voorbeeld van hoe 'n WPDV 'n klassifikasiegeval sou hanteer.

Gegee die klassifikasiegeval waar "was" geëtiketteer moet word in die sin "Hy was jou dit verskuldig." Die komplekse eienskapvektor vir die klassifikasietaak is voorgestel in (2.15).

$$\{Vorigetekseenheid = \mathrm{PDHENP}, Tekseenheid = \mathrm{was}, Volgendetekseenheid = \mathrm{PTEDP}\} \quad (2.15)$$

Eerstens lys WPDV alle komplekse en atomiese eienskapvektore wat teenwoordig is in die klassifikasiegeval. Die leë eienskapvektor {} word ook gesien as 'n eienskapvektor (Van Halteren, 1999) en daarmee saam is daar agt eienskapvektore teenwoordig. Hierdie agt eienskapvektore word in Tabel 2.4 gelys. Vir elkeen van hierdie eienskapvektore ken WPDV nou 'n waarskynlikheidsverspreiding toe. Dit beteken dat daar vir elke kandidaat 'n waarskynlikheid om 'n klassifikasie te wees toegeken word. Hierdie waarskynlikhede is vooraf beraam deur middel van die afrigtingsdata. Normaalweg word alle kandidate wat bestaan in ag geneem, maar in hierdie voorbeeld word net drie in ag geneem. Kolomme 2 tot 4 van Tabel 2.4 lys die waarskynlikhede van die drie kandidate by elke eienskapvektor. Nou word daar aan elke eienskapvektor 'n gewig toegeken soos in kolom 5 van Tabel 2.4 gesien kan word. WPDV ken groter gewigte toe aan die eienskapvektore wat meer inligting bevat, sodat dié eienskapvektore 'n groter invloed op die uiteindelike uitvoer sal uitoefen. In hierdie voorbeeld word die gewig van die eienskapvektor gebaseer op die aantal atomiese eienskapvektore waaruit die eienskapvektor saamgestel is (Van Halteren *et al.*, 2001). Die gewig is gelyk aan $N! = N \times (N-1) \times (N-2) \ldots \times 1$ waar N die aantal atomiese eienskapvektore is waaruit die

eienskapvektor saamgestel is. Daar bestaan ook ander, meer gesofistikeerde metodes om die gewig van ’n eienskapvektor te bepaal (Van Halteren, 1999), maar vir die doel van die voorbeeld word gebruik gemaak van die eenvoudiger metode van gewigbepaling. Die geweegde verspreidings van elke eienskapvektor word daarna bereken deur vir elke eienskapvektor die waarskynlikheidverspreidings met die gewig te vermenigvuldig. Vir elke eienskapvektor ontvang elke kandidaat dus nou ’n geweegde waarskynlikheid om as klassifikasie gekies te word. Elke kandidaat se geweegde waarskynlikhede word bymekaargetel en die somtotaal word in die laaste ry van kolomme 6 tot 8 van Tabel 2.4 getoon. Die kandidaat met die hoogste somtotaal, dit is VVHOK, word aangewys as die klassifikasie en die uitvoer van WPDV is $\{\text{was} = \text{VVHOK}\}$.

Eienskapvektor	Kandidaatwaarskynlikhede			Gewig	Gewig × waarskynlikhede		
	VVHOK	VVHOG	VVUOP		VVHOK	VVHOG	VVUOP
1	2	3	4	5	6	7	8
{*Vorigetekseenheid*=PDHENP, *Tekseenheid*=was, *Volgendetekseenheid*=PTEDP}	0.56	0.34	0.1	6	3.36	2.04	0.6
{*Vorigetekseenheid*=PDHENP, Tekseenheid=was}	0.62	0.02	0.36	2	1.24	0.04	0.72
{*Vorigetekseenheid*=PDHENP, *Volgendetekseenheid*=PTEDP}	0	0.01	0	2	0	0.02	0
{*Tekseenheid*=was, *Volgendetekseenheid*=PTEDP}	0.22	0.74	0.01	2	0.44	1.48	0.02
{*Vorigetekseenheid*=PDHENP}	0.02	0	0.01	1	0.02	0	0.01
{*Tekseenheid*=was}	0.65	0.2	0.12	1	0.65	0.2	0.12
{*Volgendetekseenheid*=PTEDP}	0	0.03	0	1	0	0.03	0
{}	0.02	0.01	0	0.01	0.0002	0.0001	0
Somtotaal					5.7102	3.8101	1.47

Tabel 2.4 - ’n Voorbeeld van ’n WPDV-klassifikasiegeval

Alhoewel hierdie voorbeeld ’n baie eenvoudige voorbeeld is ter verduideliking van hoe WPDV werk, bestaan daar ook baie ingewikkelder gevalle met komplekser eienskapvektore as invoer vir WPDV. Waar WPDV ’n opgestapelde sisteem is, dus ’n tweedevlakklassifiseerder, kan eienskapvektore uit die kandidate van elke eerstevlakklassifiseerder bestaan saam met die vorige en volgende kandidate. ’n Voorbeeld van ’n eienskapvektor sou dalk soos in (2.16) daar

kon uitsien waar die drie verskillende kandidate die kandidate is wat deur drie verskillende eerstevlakklassifiseerders voorgestel is.

$$\left\{\begin{array}{r} Vorigetekseenheid = \text{PDHENP+PTEDP+PEMB} \\ Tekseendheid = \text{was} \\ kandidate = \text{VVHOK+VVHOG+VVUOP} \\ Volgendetekseenheid = \text{PTEDP+PTEW+PEEDP} \end{array}\right\} \quad (2.16)$$

2.2.6 KLASSIFISEERDEROMRUILING

In hierdie metode kies die kombinasieklassifiseerder vir elke klassifikasiegeval die eerstevlak-klassifiseerder wat die meeste vertrou kan word met die spesifieke geval. Die vertrouens-waardigste eerstevlakklassifiseerder word dan aangewys om 'n klassifikasie te maak. Henderson en Brill (1998) het hierdie kombinasiemetode vir sinsontleding gebruik en dit sinsontledingomruiling (Eng. *parser switching*) genoem. Die metode behels dat kandidate in konstituente opgebreek word. In Henderson en Brill (1998) se geval sal 'n kandidaat 'n sinsontleding (Eng. *parse*) vir 'n sin wees. Daar kan aangeneem word dat die konstituente van so 'n sinsontleding uit sinstukke sal bestaan, byvoorbeeld: Sin = Naamwoordstuk + Werkwoordstuk. Elke sinstuk kan natuurlik ook verder in kleiner konstituente opgebreek word, wat uit verdere sinstukke of woordsoortetikette sal bestaan.

In Klassifiseerderomruiling word daar, net soos in die geval van die stemmetode, een van die kandidate wat deur 'n eerstevlakklassifiseerder voorgestel is, uiteindelik as die klassifikasie gekies. Die verskil lê by die metode waarop die kandidaat gekies word. Elke eerstevlak-klassifiseerder stel 'n kandidaat voor wat in konstituente opgebreek word. Elke voorgestelde kandidaat word nou met elke ander kandidaat vergelyk en die ooreenkomste word dan gemeet deurdat die konstituente wat die kandidate in gemeen het, getel word. Die eerstevlak-klassifiseerder wat die kandidaat met die hoogste somtotaal van ooreenkomste voorgestel het, word uiteindelik deur die kombinasieklassifiseerder as die klassifiseerder vir die spesifieke klassifikasiegeval aangewys. Wanneer meer as een kandidaat se somtotaal van ooreenkomste gelyk is en hierdie somtotaal die hoogste is, word daar ewekansig tussen hulle gekies.

2.2.7 VOLGORDE SISTEEM

Trushkina (2004) bou 'n klassifikasiemodel vir morfo-sintaktiese etikettering (Eng. *morpho-syntactic tagging*) wat die voordele van 'n statistiese en reëlgebaseerde model kombineer. Haar model bestaan uit 'n morfologiese analiseerder, 'n reëlgebaseerde module wat meerduidigheid verminder en 'n statistiese etiketteerder. Die volgorde van hierdie modules reflekteer die sterk-punte van elke module. In die model verskaf 'n morfologiese analiseerder eerstens alle moontlike kandidate vir 'n gegewe klassifikasietaak. Hierdie meerduidige uitvoer word dan as

invoer aan die reëlgebaseerde module gestuur. Die taak van die reëlgebaseerde module is om die hoeveelheid kandidate in die klassifikasiegeval te verminder sodat die statistiese module minder kandidate het om te oorweeg. Die reëlgebaseerde module penaliseer slegs die kandidate waarteen dit genoegsame negatiewe bewyse het. Die uitvoer van die reëlgebaseerde module is weer die invoer van die statistiese module wat nou tussen die oorblywende kandidate oordeel. Die reëlgebaseerde module tree dus op as ’n filter vir die statistiese module en het ’n verhoogde akkuraatheid vir die statistiese module tot gevolg.

2.2.8 INTERNE KOMBINASIE

Sang *et al.* (2000) pas interne kombinasie (Eng. *System Internal Combination*) toe op die klassifikasie van naamwoordstukke (Eng. *noun phrase chunking*). Interne kombinasie maak gebruik van dieselfde leeralgoritme, maar pas dit toe op variante van die taak. Die resultate word uiteindelik gekombineer deur van meerderheidstemming gebruik te maak. Die verskillende variante van die taak word gegenereer deurdat verskillende voorstellings van die uitvoer gebruik word. Vervolgens word verduidelik hoe verskillende voorstellings in naamwoordstukherkenning, wat deel uitmaak van sinstukverdeling, daar uitsien.

Sinstukverdeling (Eng. *chunking*) word dikwels as ’n etiketteringstaak gedefinieer (Ramshaw & Marcus, 1995; Veenstra, 1999), sodat daar aan elke woord binne ’n sin ’n sinstuketiket toegeken word. IOB en IOE (Sang *et al.*, 2000) is onder andere twee verskillende voorstellings vir die etikettering van naamwoordstukke. In die twee voorstellings verteenwoordig die I die begrip binne (Eng. *inside*), die O buite (Eng. *outside*), die B begin (Eng. *beginning*), en die E einde (Eng. *end*). Na aanleiding van die voorstelling wat die eerstevlakklassifiseerder gebruik, sal daar nou aan elke woord binne ’n sin een van die bogenoemde etikette (I,O,B of E) toegeken word. Gestel die invoerwoord tydens naamwoordstukverdeling is die eerste woord in ’n naamwoordstuk, dan kry die woord volgens IOB die etiket B, en volgens IOE die etiket I. Gestel die invoerwoord is glad nie deel van ’n naamwoordstuk nie, dan kry die invoerwoord volgens beide voorstellings die etiket O. Gestel die invoerwoord is deel van ’n naamwoordstuk, maar nie die eerste of laaste woord nie, dan kry die invoerwoord ook volgens beide voorstellings die etiket I. Laastens, gestel die invoerwoord is die laaste woord van ’n naamwoordstuk, dan kry die woord volgens IOB ’n I as etiket en volgens IOE ’n E as etiket. Buiten vir hierdie twee voorstellings, bestaan daar ook nog meer voorstellings vir die uitvoer van ’n sinstukverdeler (Ramshaw & Marcus, 1995; Sang & Veenstra, 1999; Veenstra, 1999).

Sang *et al.* (2000) stel die uitvoer van hulle naamwoordstukverdeler deur middel van vyf verskillende voorstellings voor. Dit beteken dat hulle vyf eerstevlakklassifiseerders met dieselfde leeralgoritme op dieselfde data afrig; die enigste verskil tussen die vyf eerstevlakklassifiseerders

is dat elk ’n ander voorstelling van die data gebruik. Nadat elke eerstevlakklassifiseerder ’n kandidaat voorgestel het, word die vyf kandidate na dieselfde voorstelling omgeskakel, waarna hulle eindelik deur middel van meerderheidstemming gekombineer word.

2.3 SAMEVATTING

In hierdie hoofstuk word drie verskillende kategorieë van masjienleer in afdeling 2.1 voorgehou, naamlik

- Gekontroleerde leer, wanneer die afrigtingsdata voorbeelde bevat wat soos die teikenprobleem lyk;
- Ongekontroleerde leer, wanneer die afrigtingsdata nie eksplisiete voorbeelde bevat wat soos die teikenprobleem lyk nie; en
- Gedeeltelik-gekontroleerde leer, wat ’n kombinasie van bogenoemde kategorieë is.

Aangesien die eksperimentele opstelling van Hoofstuk 3 en Hoofstuk 4 fokus op gekontroleerde en gedeeltelik-gekontroleerde leer, word daar in meer detail op hierdie twee kategorieë gefokus. Vir gekontroleerde leer word klassifikasie in afdeling 2.1.1.1 bespreek met twee van die bekendste algoritmes, naamlik

- Die naïewe Bayes-algoritme; en
- *k*-Naastebuurpuntalgoritme.

In afdeling 2.1.3 word ’n paar van die belangrikste gedeeltelik-gekontroleerde algoritmes uit die literatuur bespreek aan die hand van die werking van die algoritme, voorbeelde van areas waar dit gebruik word en die voor- en nadele van die gebruik van die algoritme. Die algoritmes is

- Selfleer;
- Deelleer;
- Generatiewe modelle;
- Gedeeltelik-gekontroleerde Steunvektorklassifiseerders; en
- Grafiekgebaseerde metodes.

In die tweede deel van die hoofstuk in afdeling 2.2 word die tegniek van die kombinering van verskeie algoritmes vir die klassifikasietaak bespreek. Die belangrikste kombinasiemetodes uit die literatuur word aan die hand van die werking van die metode en die sukses daarvan vir taaltegnologiese toepassings bespreek. Die metodes wat bespreek word, is

- Kaskademetode;
- Stemmetode;
- Versterking;
- Groepering;
- Opstapeling;
- Klassifiseerderomruiling;
- Volgorde sisteem; en
- Interne kombinasie

In die volgende hoofstuk word die stemmetode (afdeling 2.2.2) as kombinasietegniek aangewend om ’n kombinasie-woordsoortetiketteerder vir Afrikaans saam te stel.

Hoofstuk 3

’N AFRIKAANSE KOMBINASIE-WOORDSOORTETIKETTEERDER

3.1 METODE

Die voorgestelde Afrikaanse kombinasie-woordsoortetiketteerder kombineer vier verskillende gekontroleerde leerklassifiseerders wat afgerig is vir die taak van woordsoortetikettering. Die eerstevlakklassifiseerders se uitsette word gekombineer deur middel van die stemmetode (afdeling 2.2.2) en bied verskeie opsies aan die gebruiker met betrekking tot die spesifieke stemmetode wat gebruik moet word en die eerstevlakklassifiseerders wat ingesluit moet wees. Dieselfde afrigtingsdata wat gebruik is vir die gedeeltelik-gekontroleerde metode van Hoofstuk 4 word ook gebruik vir die kombinasie-woordsoortetiketteerder met die weglating van die ongeëtiketteerde data (afdeling 4.3), naamlik 30000 korrek geëtiketteerde tekseenhede soos verskaf deur die Sentrum vir Tekstegnologie (CTexT®) van die Noordwes-Universiteit. Die data is ewekansig verdeel in 80% afrigtingsdata, 10% toetsdata (voortaan TEST) en 10% insteldata (Eng. *tune data*). Verder word daar vir die doel van hierdie eksperimente nog ’n stel toetsdata bygevoeg soos ook gebruik deur Pilon (2005) naamlik die Afrikaanse weergawe van die Universele Verklaring van Menseregte (voortaan UVVM).

3.2 EERSTEVLAKKLASSIFISEERDERS

Vervolgens word ’n beskrywing van die gekontroleerde leeralgoritmes waarop die onderskeie eerstevlakklassifiseerders gebaseer is, weergegee. Die eerstevlakklassifiseerders word elk individueel met behulp van beskikbare pakkette afgerig op dieselfde afrigtings- en toetsdata, waarna daar vir elke klassifiseerder maatstawwe bereken word wat deur die kombinasie-woordsoortetiketteerder tydens etikettering gebruik word.

3.2.1 MBT

MBT[2] (Daelemans *et al.*, 1996) is ’n geheuegebaseerde woordsoortetiketteerder (Eng. *Memory-based tagger generation and tagging*) wat gebaseer is op TiMBL (Eng. *Tilburg Memory-Based Learner*). TiMBL (Daelemans *et al.*, 2003) implementeer verskeie geheuegebaseerde leer-algoritmes wat elk ’n uitdruklike weergawe van die afrigtingsdata in die geheue stoor. Gedurende klassifikasie word nuwe gevalle geëkstrapoleer vanaf die mees soortgelyke gestoorde gevalle. Die algoritmes wat in TiMBL gebruik word, verskil in drie opsigte, naamlik die

[2] MBT kan afgelaai word by http://ilk.uvt.nl/mbt/

definisie van gelyksoortigheid; die manier waarop die voorbeelde in die geheue gestoor word; en die manier waarop die soektog deur die geheue gelei word.

Elke voorbeeld, of te wel klassifikasiegeval, word voorgestel deur ’n eienskapvektor met ’n vasgestelde lengte. Indien die klassifikasiegeval voorheen waargeneem is, dus in die gestoorde klassifikasiegevalle, of te wel *klassifikasiegevalbasis* (Eng. *case base*) voorkom, word die mees frekwente ooreenkomstige klassifikasie toegeken. Indien die klassifikasiegeval nie voorkom onder die gestoorde klassifikasiegevalle nie, word die k naaste buurpunte bepaal met behulp van ’n gelyksoortigheidsmetriek, waarna die klassifikasie gebaseer word op die waargenome klassifikasies vir daardie buurpunte.

Wanneer daar ’n geëtiketteerde korpus beskikbaar is, behels die woordsoortetiketteerder-generering deur middel van MBT die outomatiese onttrekking van ’n leksikon, ’n klassifikasie-gevalbasis vir bekende woorde en ’n klassifikasiegevalbasis vir onbekende woorde. Die leksikon assosieer woorde met dubbelsinnigheidsklasse van etikette soos wat dit uit die afrigtingsdata waargeneem is. Die klassifikasiegevalbasisse word in ’n heuristies geïndekseerde weergawe van ’n klassifikasiegevalgeheue genaamd IGTree (Daelemans et al., 1997) gestoor. Die klassifikasiegevalle word voorgestel deur ’n verskeidenheid eienskapvektore waarvan die relevansie outomaties deur inligtingswinsgewigte (Eng. *information gain*) bepaal word.

Gedurende etikettering word nuwe klassifikasiegevalle in die volgorde van die belangrikste eienskapvektor tot by die onbelangrikste eienskapvektor met klassifikasiegevalle in die geheue vergelyk. Dit behels dat elke tekseenheid wat geëtiketteer moet word in die leksikon opgesoek word en, indien die tekseenheid in die leksikon voorkom, word die leksikale voorstelling onttrek en die konteks bepaal. Dubbelsinnigheid word verwyder deur te ekstrapoleer vanaf naaste buurpunte in die klassifikasiegevalbasis vir bekende woorde. Wanneer ’n tekseenheid nie voorkom in die leksikon nie, word sy leksikale voorstelling bereken op grond van sy vorm waarna sy konteks bepaal word en dubbelsinnigheid verwyder word deur te ekstrapoleer vanaf naaste buurpunte in die klassifikasiegevalbasis vir onbekende woorde. Vir elke klassifikasie-geval sal die uitvoer die beste raaiskoot van die kategorie vir die tekseenheid in sy huidige konteks wees.

3.2.2 SVM^{LIGHT}

SVM^{light} (Joachims, 1999a) is ’n implementering van steunvektorklassifiseerders wat aanvanklik deur Vapnik (1998) voorgestel is en wat in afdeling 2.1.3.4 in meer detail bespreek word[3].

[3] SVM^{light} kan afgelaai word by http://svmlight.joachims.org/

Figuur 2.2 op bladsy 24 illustreer ’n voorbeeld van ’n geëtiketteerde datastel wat deur middel van steunvektorklassifiseerders voorgestel is.

SVM*light* gebruik verskeie optimiseringsalgoritmes en beskik oor skaalbare geheuevereistes wat dit in staat stel om groot probleme met duisende steunvektore en honderde duisende afrigtings-voorbeelde effektief te hanteer. Dit maak gebruik van yl voorkomsvektore, sluit effektiewe algoritmes in om die foutkoers, presisie en herroeping te bereken en kan op ’n groot verskeidenheid probleemtipes toegepas word, onder andere teksklassifikasie, beeldherkenning en mediese toepassings.

3.2.3 MXPOST

MXPOST[4] (Ratnaparkhi, 1996) gebruik eienskapvektore uit die konteks om ’n woordsoortetiket te voorspel deur middel van maksimum entropie en is ontwerp met die doel om afrigbaar te wees op enige groot korpus wat geëtiketteer is met woordsoortetikette. Die waarskynlikheidsmodel is gedefinieer oor $H \times T$ waar H die versameling moontlike tekseenhede met etikette uit die afrigtingsdata (Eng. *histories*) is, terwyl T die versameling toelaatbare etikette verteenwoordig. Die waarskynlikheidsverdeling $p(h,t)$, waar h die invoer en t die etikette is, word gekies om die beste passing op die data te vind. Die maksimum aanneemlikheid van die data volgens die waarskynlikheidsmodel word verkry deur die parameterwaardes θ te kies waarvoor die data die hoogste aanneemlikheid het, soos voorgestel in (3.1).

$$max_{\theta} p(\theta | data) = max_{\theta} p(\theta) \times p(data | \theta) \qquad (3.1)$$

Die maksimum aanneemlikheidsmodel kan ook geïnterpreteer word as die maksimering van entropie onderhewig aan sekere beperkings, waar entropie gegee word deur (3.2).

$$G(p) = - \sum_{h \in H, t \in T} p(h,t) \, log \, p(h,t) \qquad (3.2)$$

Gegee (h,t), kan ’n eienskapvektor geaktiveer word op enige tekseenheid of etiket in h, en moet die eienskapvektor enige inligting enkodeer wat kan help om t te voorspel, soos byvoorbeeld die spelling van die tekseenheid of die vorige twee etikette.

---

[4] MXPOST kan afgelaai word by ftp://ftp.cis.upenn.edu/pub/adwait/jmx/

Die generering van eienskapvektore vir die etikettering van onbekende woorde is gebaseer op die hipotese dat skaars woorde in die afrigtingsdata ooreenkomste het met onbekende woorde uit die toetsdata met betrekking tot hoe spelling help om die etikette te voorspel. Dus sal skaars woordeienskapvektore wat gegenereer is uit die afrigtingsdata toegepas word op skaars woorde, sowel as onbekende woorde in die toetsdata. Die gedrag van ’n eienskapvektor wat selde voorkom in die afrigtingsdata is dikwels moeilik om te voorspel, aangesien sy statistiek nie betroubaar is nie. Om hierdie rede neem die model aan dat enige eienskapvektor wat minder as 10 keer in die afrigtingsdata voorgekom het onbetroubaar is en word hierdie eienskapvektore geïgnoreer.

3.2.4 TNT

TnT[5] (Eng. *Trigrams'n'Tags*) is ’n gekontroleerde leer, statistiese woordsoortetiketteerder wat die Viterbi-algoritme van tweede orde Markov-modelle implementeer (Brants, 2000). Dit is ontwikkel met die doel om afrigbaar te wees op ’n verskeidenheid tale en etiketstelle en is geoptimeer vir spoed en afrigting op ’n groot verskeidenheid korpusse. TnT gebruik die metode van suffiksanalise vir die hantering van onbekende woorde. Die waarskynlikheidsverdeling vir ’n spesifieke suffiks word afgelei uit al die tekseenhede in die afrigtingsdata met dieselfde suffiks. ’n Suffiks is die laaste string karakters van ’n tekseenheid met ’n sekere voorafbepaalde lengte en nie noodwendig ’n linguisties sinvolle suffiks nie. Die waarskynlikhede word gladgestryk deur middel van opeenvolgende abstraksie (Eng. *successive abstraction*) wat die waarskynlikheid van ’n etiket bereken deur opeenvolgend meer en meer karakters van die suffiks weg te laat. Die maksimumaanneemlikheidsberamer vir ’n suffiks word afgelei vanuit die afrigtingsdata-frekwensies.

Soos uit die naam afgelei word, maak TnT gebruik van trigramme, wat ’n groep opeenvolgende tekseenhede is, om klassifikasies toe te ken. In vergelyking met die bigram wat die vorige woord gebruik, gebruik ’n trigram die vorige twee woorde en dus die N-gram die vorige $N-1$ woorde. ’n Bigram word ’n eerste orde Markov-model genoem, terwyl ’n trigram ’n tweede orde Markov-model en ’n N-gram ’n $N-1$ orde Markov-model genoem word (Jurafsky & Martin, 2000). TnT se gebruik van trigramme behels dat dit die vorige twee woordsoortetikette eerder as die tekseenheid self as eienskapvektore gebruik om die huidige tekseenheid te klassifiseer. Aangesien TnT ontwikkel is met die doel om afrigbaar te wees op ’n verskeidenheid tale en etiketstelle

[5] TnT kan verkry word deur Thorsten Brants te kontak en ‘n lisensie-ooreenkoms te onderteken. Hunpos is ‘n “oop bron” gelisensieerde herimplementasie van TnT wat beskikbaar is by https://code.google.com/p/hunpos/

bestaan daar egter gewoonlik nie genoeg voorbeelde van elke trigram om ’n betroubare waarskynlikheid te beraam nie. Om hierdie rede is die paradigma wat TnT gebruik vir gladstryking (Eng. *smoothing*) ’n konteks-onafhanklike variant van liniêre interpolasie met waardes vir gewigte wat nie afhanklik is van ’n spesifieke trigram nie. As gevolg van die dataskaarsheidsprobleem is dit nie moontlik om ’n aparte stel gewigte vir elke trigram te beraam nie en daarom word trigramme volgens frekwensie gegroepeer en saamgevoegde stelle van gewigte beraam deur middel van geskrapte interpolasie (Eng. *deleted interpolation*). Dit behels die verwydering van elke trigram van die afrigtingsdata en die beraming van die beste waardes vir die gewigte vanuit alle ander N-gramme in die afrigtingsdata.

3.3 KOMBINASIEKLASSIFISEERDER

Die kombinasie-woordsoortetiketteerder maak gebruik van geweegde stemming waarvolgens gewigte vir die onderskeie eerstevlakklassifiseerders bereken word deur gebruik te maak van presisie en herroeping (afdeling 1.6.3).

Vir elke klassifikasiegeval kry alle eerstevlakklassifiseerders elk ’n stem vir die etiket van sy keuse sodat die etiket met die hoogste aantal stemme toegeken word. Die kombinasie-klassifiseerder bied verskeie verstellings wat bepaal hoe groot die invloed van elke eerstevlak-klassifiseerder se stem sal wees (Van Halteren *et al.*, 2001). Die mees demokratiese opsie sou wees om meerderheidstemming toe te pas deur aan elke eerstevlakklassifiseerder ’n stem van gelyke gewig toe te ken. Dit maak egter meer sin om elke eerstevlakklassifiseerder se stem te vermenigvuldig met ’n gewig wat verband hou met hoe goed daardie klassifiseerder gevaar het in toetsing gedurende die afrigtingsfase. Vir hierdie doel word presisie en herroeping gebruik om die gewigte te bepaal. Wanneer daar op ’n oorhoofse vlak gemeet word hoeveel tekseenhede in totaal korrek geëtiketteer is en dit geld, soos in hierdie geval, dat daar slegs een etiket per teks-eenheid toegeken word deur elke eerstevlakklassifiseerder, sal presisie en herroeping gelyk wees en kan dit ook as akkuraatheid uitgedruk word. Hierdie spesifieke stemmetode wat die oorhoofse of totale presisie van elke eerstevlakklassifiseerder as gewig toeken, word *Totale-Presisie* genoem.

Dit is egter moontlik om gewigte op ’n meer gedetailleerde vlak toe te ken wat van toepassing is op die huidige situasie, soos in die volgende twee stemmetodes verduidelik word.

Volgens die *Etiket-Presisie* stemmetode stem elke eerstevlakklassifiseerder met ’n gewig wat gelyk is aan die presisie wat die betrokke klassifiseerder vir die voorgestelde etiket gedurende die afrigtingsfase behaal het.

Die *Presisie-Herroeping* stemmetode ken ’n gewig aan elke eerstevlakklassifiseerder toe wat gebaseer is op hoe goed die klassifiseerder vaar met die voorgestelde etiket (presisie), asook

hoe gereeld die klassifiseerder sleg vaar met die etikette wat deur die opposisie-eerstevlak-klassifiseerders (herroeping) voorgestel is.

Vervolgens word ’n uiteensetting gegee van die sewe verskillende verstellings wat die Afrikaanse kombinasie-woordsoortetiketteerder bied. Hierdie opsies is elkeen gebaseer op een van die drie stemmetodes soos hierbo verduidelik. Alle opsies maak gebruik van geweegde stemming. Die etiketteerder bied die opsie om enige van die eerstevlakklassifiseerders buite rekening te laat. Uit die aard van die saak is dit noodsaaklik om ’n minimum van een eerstevlak-klassifiseerder te kies. Die verstekopsie van die etiketteerder laat al vier eerstevlak-klassifiseerders deelneem aan die stemproses.

Presisie-Herroeping metode – Opsie 1

```
vir die voorgestelde etiket
    as etiket presisie > 0.5 en gemiddeld(herroeping) > 0.5
        stel gewig = etiket presisie + (1-gemiddeld(herroeping))
    anders
        stel gewig = etiket presisie
```

Figuur 3.1 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 1

Presisie-Herroeping metode – Opsie 2

Hierdie is die verstekopsie wat deur die etiketteerder toegepas word.

```
vir die voorgestelde etiket
    as etiket presisie > 0.5 en maksimum(herroeping) > 0.5
        stel gewig = etiket presisie + (1-maksimum(herroeping))
    anders
        stel gewig = etiket presisie
```

Figuur 3.2 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 2

Presisie-Herroeping metode – Opsie 3

```
vir die voorgestelde etiket
    as etiket presisie > 0 en gemiddeld(herroeping) > 0
        stel gewig = etiket presisie + (1-gemiddeld(herroeping))
    anders
        stel gewig = etiket presisie
```

Figuur 3.3 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 3

Presisie-Herroeping metode – Opsie 4

```
vir die voorgestelde etiket
    as etiket presisie > 0.2 en gemiddeld(herroeping) > 0.2
        stel gewig = etiket presisie + (1-gemiddeld(herroeping))
    anders
        stel gewig = etiket presisie
```

Figuur 3.4 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 4

Presisie-Herroeping metode – Opsie 5

```
vir die voorgestelde etiket
    as etiket presisie > 0 en maksimum(herroeping) > 0
        stel gewig = etiket presisie + (1- maksimum(herroeping))
    anders
        stel gewig = etiket presisie
```

Figuur 3.5 – Pseudokode vir die berekening van die gewig vir elke eerstevlakklassifiseerder volgens Opsie 5

Opsie 1 tot Opsie 5 in Figuur 3.1 tot Figuur 3.5 maak gebruik van die Presisie-Herroeping stem-metode waar die gewig wat toegeken word aan elke eerstevlakklassifiseerder gelyk is aan $Presisie + (1 - Herroeping)$. Ter verduideliking: gestel Klassifiseerder A stel $kandidaat_x$ voor en Klassifiseerder B stel $kandidaat_y$ voor. Die stemgewigte vir die onderskeie kandidate word dan bereken soos uiteengesit in Tabel 3.1, en $kandidaat_y$ word toegeken, aangesien dit in hierdie voorbeeld die grootste stemgewig het.

Klassifiseerder A		Klassifiseerder B	
Presisie $kandidaat_x$	Herroeping $kandidaat_y$	Presisie $kandidaat_y$	Herroeping $kandidaat_x$
0.9658	0.8927	0.9044	0.9767
$kandidaat_x$ stemgewig: $0.9658 + 1 - 0.9767 = 0.9891$			
$kandidaat_y$ stemgewig: $0.9044 + 1 - 0.8927 = 1.0117$			

Tabel 3.1 – Presisie-Herroeping stemmetode voorbeeld

Indien die gewigte van al die eerstevlakklassifiseerders almal gelyk is aan 0, word die totale presisie van elke klassifiseerder as gewig toegeken. Die herroeping van 'n ander eerstevlak-klassifiseerder word net in ag geneem indien die voorgestelde etiket verskil van die voorstel wat die klassifiseerder ter sprake gemaak het. Waar daar meer as een herroeping in ag geneem moet word, byvoorbeeld waar klassifiseerder A $kandidaat_a$ voorstel en klassifiseerder B stel $kandidaat_b$ voor en klassifiseerder C stel $kandidaat_c$ voor, dan moet klassifiseerder A $herroeping_b$ en $herroeping_c$ in berekening bring. Dit word gedoen deur die gemiddeld van $herroeping_b$ en $herroeping_c$ te neem of slegs die grootste herroeping te neem en die ander te ignoreer.

Etiket-Presisie metode – Opsie 6

Die Etiket-Presisie stemmetode word in hierdie opsie gebruik waar die gewig vir elke eerstevlak-klassifiseerder gelyk gestel word aan die etiketpresisie van die voorgestelde etiket.

Totale-Presisie metode – Opsie 7

Die geweegde stemming wat in hierdie opsie gebruik word, pas die Totale-Presisie stemmetode toe waar die gewig vir elke eerstevlakklassifiseerder gelyk gestel word aan die totale presisie van die betrokke klassifiseerder.

3.4 RESULTATE

Die resultate behels eerstens net die akkuraatheid van die vier eerstevlakklassifiseerders waarna die ooreenkomste tussen die eerstevlakklassifiseerders en die potensiële verbetering wat die kombinasie van die klassifiseerders kan meebring, ondersoek word. Laastens word die werklike resultate van kombinasie weergegee.

3.4.1 KWALITEIT VAN EERSTEVLAKKLASSIFISEERDERS

Die akkuraatheid (%) van die vier eerstevlakklassifiseerders op die toetsdata TEST en UVVM word in Tabel 3.2 weergee. Die moeilikheidsgraad van UVVM blyk hoër te wees as dié van TEST, aangesien die akkuraatheid van UVVM deurgaans laer is as dié van TEST. Dit kan wees omdat UVVM nie soos TEST ewekansig uit die afrigtingsdata onttrek is nie en dus nie in dieselfde mate as TEST verteenwoordigend is van die afrigtingsdata nie. Daar kom meer gevalle in UVVM voor wat glad nie in die afrigtingsdata voorgekom het nie en dus heeltemal nuut is vir die klassifiseerders. Die volgorde van beste presteerder tot swakste is vir albei toets-

datastelle dieselfde met TnT wat die beste vaar, gevolg deur MBT en MXPOST. SVM*light* vaar vir beide toetsdatastelle die swakste.

Eerstevlakklassifiseerder	%	
	UVVM	TEST
MBT	77.93%	85.87%
SVMlight	73.99%	82.38%
MXPOST	77.87%	85.37%
TnT	80.80%	89.42%

Tabel 3.2 – Akkuraatheid van eerstevlakklassifiseerders op UVVM en TEST

3.4.2 Ooreenkomste tussen eerstevlakklassifiseerders

Die uitvoerbaarheid van kombinasie kan bepaal word deur die individuele eerstevlak-klassifiseerders se uitvoer te bestudeer, aangesien kombinasie afhanklik is van verskillende sisteme wat verskillende foute produseer. Die patrone van ooreenkomste of verskille in uitvoer tussen die vier klassifiseerders op die toetsdatastelle UVVM en TEST word in Tabel 3.3 gegee. Vir elke patroon word die persentasie waarvoor die patroon waargeneem is (%), sowel as die kumulatiewe persentasie (%Kum) gegee.

Patroon	UVVM		TEST	
	%	%Kum	%	%Kum
Alle etiketteerders stem saam en is korrek	62.89%	62.89%	72.34%	72.34%
Meerderheid (maar nie alle) etiketteerders korrek	16.39%	79.28%	15.47%	87.81%
Korrekte etiket teenwoordig, maar etiketteerders is gelykop	2.70%	81.98%	2.81%	90.62%
Minderheid etiketteerders korrek	6.76%	88.74%	3.65%	94.27%
Etiketteerders verkil, maar almal is verkeerd	3.83%	92.57%	2.78%	97.05%
Etiketteerders stem saam, maar almal is verkeerd	7.43%	100.00%	2.95%	100.00%

Tabel 3.3 – Patrone van ooreenkomste tussen die vier klassifiseerders op UVVM en TEST

Daar kan gesien word dat die UVVM-toetsdatastel inderdaad moeiliker is, aangesien $7.43\% + 3.83\% = 11.26\%$ van die klassifikasiegevalle deur alle eerstevlakklassifiseerders verkeerd geklassifiseer is teenoor slegs 2.78% + 2.95% = 5.73% in die geval van TEST. Vir die metode van kombinasie wat in hierdie studie gebruik is, waar die uiteindelike klassifikasie wat toegeken word altyd ’n klassifikasie sal wees wat deur ten minste een eerstevlakklassifiseerder

geklassifiseer is, bestaan daar dus geen moontlikheid vir ’n kombinasieklassifiseerder om ’n foutkoers van minder as 11.26% (akkuraatheid 88.74%) op UVVM te bereik nie. Op TEST is die foutkoers nagenoeg die helfte minder met $2.95\% + 2.78\% = 5.73\%$ (akkuraatheid 94.27%). Die waarde van kombinasie lê in die moontlikheid om in gevalle waar die patroonvoorkomste die korrekte klassifikasie in meerdere of mindere mate wel insluit, die korrekte klassifikasie te voorspel. Vir UVVM beteken dit $16.39\% + 2.70\% + 6.76\% = 25.85\%$ van die toetsdata en vir TEST is dit $15.47\% + 2.81\% + 3.65\% = 21.93\%$ waarvan 6.76% en 3.65% onderskeidelik baie moeilik sal wees om reg te klassifiseer, aangesien die minderheid etikette in daardie gevalle korrek is.

3.4.3 DIE TEORIE VAN KOMBINASIE

Tabel 3.4 lys ’n paar teoretiese kombineerders wat uit die vier eerstevlakklassifiseerders saamgestel is. Die akkuraatheid (%), asook die foutkoersreduksie met betrekking tot die beste eerstevlakklassifiseerder (Δ_{Err}) vir TEST en UVVM word gegee.

Patroon	UVVM		TEST	
	%	Δ_{Err}	%	Δ_{Err}
Beste Eerstevlakklassifiseerder	80.80%	-	89.42%	-
Alle en meerderheid reg	79.28%	-7.92%	87.81%	-15.19%
Gelykop ewekansig reg	80.67%	-0.66%	89.20%	-2.06%
Alle gelykop reg	81.98%	6.16%	90.62%	11.39%
Minderheid teenoor 2 reg	84.91%	21.41%	91.66%	21.20%
Minderheid teenoor 3 reg	88.74%	41.35%	94.27%	45.89%

Tabel 3.4 – Beraamde akkuraathede van toenemende vlakke van sukses

Daar word aangeneem dat die kombineerders altyd die regte klassifikasie toeken in die gevalle waar alle of die meerderheid van die eerstevlakklassifiseerders ’n korrekte kandidaat voorgestel het. In Tabel 3.4 kan gesien word dat hierdie meerderheid nog nie voldoende is om ’n foutkoers-reduksie vir enige van die toetsdatastelle teweeg te bring nie. Die kombineerders se sukses met gelykop gevalle of selfs gevalle waar die minderheid korrek is, sal bepaal of dit teoreties moontlik is om deur kombinasie ’n foutkoersreduksie teweeg te bring. Gestel dat vir gelykop gevalle die kombineerder ewekansig uit die kandidate ’n klassifikasie toeken, dan word die regte klassifikasie 50% van die tyd vir ’n (2-2) gelykop toegeken of 25% vir ’n (1-1-1-1) gelykop of 33.33% vir ’n (1-1-1) gelykop toegeken. Soos gesien in Tabel 3.4 het hierdie ewekansige toekenning van die korrekte klassifikasie steeds nie ’n foutkoersreduksie tot gevolg nie. Eers wanneer daar aangeneem word dat die kombineerder vir gelykopgevalle altyd die regte

kandidaat sal kies, is daar ’n foutkoersreduksie van 6.16% op UVVM en 11.39% op TEST. As aangeneem word dat die kombineerder selfs in gevalle waar daar ’n minderheid teenoor 2 of 3 is, steeds die korrekte kandidaat sal toeken, is die maksimum foutkoersreduksie met betrekking tot die beste eerstevlakklassifiseerder 41.35% op UVVM en 45.89% op TEST. Die resultate van hierdie teoretiese kombineerders bring nie goeie vooruitsigte vir ’n kombinasieklassifiseerder in hierdie geval nie. Soos gesien in Tabel 3.2 vaar TnT in vergelyking met die ander drie eerstevlakklassifiseerders aansienlik beter. Gevolglik word die uitvoerbaarheid van ’n teoretiese kombineerder op die drie eerstevlakklassifiseerders wat swakker gevaar het, getoets. Tabel 3.5 toon ’n groter foutkoersreduksie met betrekking tot die beste eerstevlakklassifiseerder op alle vlakke vir beide UVVM en TEST.

Patroon	UVVM - TnT		TEST - TnT	
	%	Δ_{Err}	%	Δ_{Err}
Beste Eerstevlakklassifiseerder	77.93%	-	85.87%	-
Alle en meerderheid reg	78.72%	4.11%	87.84%	18.67%
Gelykop ewekansig reg	80.29%	12.32%	88.81%	27.85%
Alle gelykop reg	82.26%	22.58%	89.52%	34.49%
Minderheid teenoor 2 reg	87.05%	47.51%	92.80%	65.51%

Tabel 3.5 – Beraamde akkuraathede van toenemende vlakke van sukses - TnT uitgesluit

Uit hierdie resultate kan afgelei word dat vir kombinasie om suksesvol te wees die akkuraatheid van die eerstevlakkombineerders nie te veel moet verskil nie, aangesien die moontlikheid bestaan dat swakker klassifiseerders die sterker klassifiseerders kan oorweldig. Hierdie teoretiese resultate neem ook wel nog nie die individuele sterkpunte, byvoorbeeld etiketpresisie, van die onderskeie eerstevlakklassifiseerders in ag nie.

3.4.4 KOMBINASIE IN DETAIL

Die resultate vir spesifieke tekseenhede en woordsoortetikette word vervolgens bespreek. Die resultate van die eerstevlakklassifiseerders word telkens gegee, tesame met dié van ’n kombinasieklassifiseerder. Die kombinasieklassifiseerder wat gebruik word, is gebaseer op die Presisie-Herroeping – Opsie 2 stemmetode (voortaan Kombinasie2). Tabel 3.6 gee ’n lys van tekseenhede wat dikwels verkeerd geëtiketteer word. Vir elke tekseenheid word die aantal voorkomste in die toetsdatastel TEST met die aantal verskillende etikette wat met die tekseenheid geassosieer word (n/etikette) gegee. Verder word die aantal foute wat gemaak is (# foute) en die persentasie van die voorkomste wat verkeerd geëtiketteer is (%) vir elke eerstevlakklassifiseerder, sowel as vir Kombinasie2 gegee. Die laaste ry gee vir die teks-

eenhede wat gelys is in Tabel 3.6 die totaal van n/etikette en #foute en die gemiddeld oor al die voorkomste wat verkeerd geëtiketteer is.

Teks-eenheid	n / etikette	MBT		SVM^{light}		TnT		MXPOST		Kombinasie 2	
		# foute	%	# foute	%	# foute	%	# foute	%	# foute	%
my	16/3	8	50.00%	7	43.75%	4	25.00%	9	56.25%	8	50.00%
is	46/3	8	17.39%	8	17.39%	6	13.04%	6	13.04%	7	15.22%
"	30/3	6	20.00%	10	33.33%	3	10.00%	4	13.33%	3	10.00%
ander	7/3	3	42.86%	3	42.86%	1	14.29%	3	42.86%	2	28.57%
so	7/3	2	28.57%	3	42.86%	2	28.57%	6	85.71%	2	28.57%
haar	5/3	2	40.00%	2	40.00%	2	40.00%	2	40.00%	2	40.00%
En	13/2	3	23.08%	5	38.46%	4	30.77%	4	30.77%	5	38.46%
te	34/2	2	5.88%	3	8.82%	2	5.88%	3	8.82%	3	8.82%
jou	19/2	1	5.26%	1	5.26%	0	0.00%	3	15.79%	3	15.79%
nie	37/2	3	8.11%	12	32.43%	3	8.11%	1	2.70%	2	5.41%
waar	8/2	3	37.50%	2	25.00%	2	25.00%	2	25.00%	2	25.00%
gaan	9/2	3	33.33%	3	33.33%	0	0.00%	0	0.00%	0	0.00%
hulle	9/2	0	0.00%	1	11.11%	1	11.11%	8	88.89%	0	0.00%
Totaal	**240/32**	**44**	**24.00%**	**60**	**28.82%**	**30**	**16.29%**	**51**	**32.55%**	**39**	**20.45%**

Tabel 3.6 – Tekseenhede in TEST wat dikwels verkeerd geëtiketteer is

In die aantal foute vir elke klassifiseerder kan dieselfde tendens soos wat reeds in die vorige tabelle opgemerk is, gesien word met die aantal foute wat vir die beste presteerder TnT in die meeste gevalle minder is as die aantal foute vir Kombinasie2. Vir die ingewikkelde tekseenhede vaar Kombinasie2 dus nie beter as TnT nie, maar dit verhoog eerder die foutpersentasie. Kombinasie2 toon wel ’n verbetering vir die meeste van die tekseenhede in vergelyking met die oorblywende drie eerstevlakklassifiseerders. Vir die laaste ry (Totaal) het TnT ook die minste totale aantal foute (30) en die kleinste gemiddeld oor al die voorkomste (16.29%), gevolg deur Kombinasie2 (39 en 20.45%) en MBT (44 en 24.00%). Aan die ander kant is die totale aantal foute wat gemaak is deur MXPOST (51) wel minder as SVM*light* (60), maar die gemiddeld van SVM*light* (28.82%) is laer as dié van MXPOST (32.55%) wat ’n aanduiding gee dat SVM*light* dalk beter vaar met skaars tekseenhede.

Tabel 3.7 gee etiketpare wat dikwels met mekaar verwar word. Vir elke paar (Klassifiseerder, Korrek) word die twee verwarringsrigtings apart gelys met die aantal foute wat deur elke eerstevlakklassifiseerder, asook Kombinasie2 in TEST gemaak is. Daarna word die aantal foute wat vir elke paar as geheel gemaak is, aldus die twee verwarringsrigtings, saam gelys.

Klassifiseerder	Korrek	MBT	SVM^{light}	MXPOST	TnT	Kombinasie2
VTHOG	VTHOO	15	15	14	20	14
VTHOO	VTHOG	13	9	7	10	9
	paar	28	24	21	**30**	23
VVHOG	VVHOO	12	13	4	12	13
VVHOO	VVHOG	11	9	11	10	10
	paar	**23**	22	15	22	23
NSE	NM	10	14	16	9	12
NM	NSE	13	4	7	11	5
	paar	23	18	23	20	17
NEE	RVE	7	10	13	10	10
RVE	NEE	1	0	1	1	1
	paar	8	10	14	11	11
PEEB	PEEDP	6	6	7	1	7
PEEDP	PEEB	0	0	0	0	0
	paar	6	6	7	1	7

Tabel 3.7 – Etiketpare wat dikwels met mekaar verwar word met die aantal foutvoorkomste

Vir die $\mathrm{VTHOG} \leftrightarrow \mathrm{VTHOO}$ paar waar VTHOG ’n oorganklike werkwoord is en VTHOO ’n onoorganklike werkwoord is (kyk Bylaag B vir volledige beskrywings van etikette), is Kombinasie2 wel voordelig vir die toppresteerder TnT waar TnT ’n totaal van 30 foutvoorkomste het teenoor Kombinasie2 se 23 foutvoorkomste. Verder vind TnT ook baat by Kombinasie2 vir die paar $\mathrm{NSE} \leftrightarrow \mathrm{NM}$, terwyl dit geen invloed het vir $\mathrm{NEE} \leftrightarrow \mathrm{RVE}$ nie. Soos in vorige resultate word TnT hier ook in sommige gevalle afgestem tot die punt waar die kwaliteit nie meer behoue bly nie, soos vir pare $\mathrm{VVHOG} \leftrightarrow \mathrm{VVHOO}$ en $\mathrm{PEEB} \leftrightarrow \mathrm{PEEDP}$. Dit is interessant om op te merk dat vir die swakste eerstevlakklassifiseerder (SVM^{light}) Kombinasie2 in die drie pare $\mathrm{VVHOG} \leftrightarrow \mathrm{VVHOO}$, $\mathrm{NEE} \leftrightarrow \mathrm{RVE}$ en $\mathrm{PEEB} \leftrightarrow \mathrm{PEEDP}$ nie tot voordeel is nie. Dit blyk dus dat SVM^{light} in vergelyking met die ander eerstevlakklassifiseerders redelik goed vaar met etikette wat dikwels verwarring by eerstevlakklassifiseerders veroorsaak. Dit sluit aan by die waarneming in Tabel 3.6 waar SVM^{light} in vergelyking beter vaar met tekseenhede wat dikwels verwarring veroorsaak.

Tabel 3.8 gee vir elke etiket wat in Tabel 3.7 voorkom die persentasie wat die etiket in die toetsdatastel TEST uitmaak (% in TEST) asook die presisie en herroeping vir elke eerstevlakklassifiseerder en Kombinasie2.

Etiket	% in TEST	MBT		SVM^{light}		MXPOST		TnT		Kombinasie2	
		Presisie	Her-roeping	Presisie	Her-roeping	Presisie	Her-roeping	Presisie	Her-roeping	Presisie	Her-roeping
NEE	5.73%	89.41%	88.89%	81.00%	**94.74%**	82.14%	94.15%	**89.83%**	92.98%	88.04%	**94.74%**
NM	1.81%	55.77%	53.70%	**75.00%**	50.00%	68.29%	51.85%	66.67%	**62.96%**	73.17%	55.56%
NSE	9.34%	83.33%	82.44%	75.38%	87.81%	80.78%	88.89%	**88.30%**	89.25%	86.94%	**90.68%**
PEEB	0.20%	41.67%	83.33%	46.15%	**100.00%**	38.46%	83.33%	**85.71%**	**100.00%**	42.86%	**100.00%**
PEEDP	0.30%	**100.00%**	33.33%	**100.00%**	33.33%	**100.00%**	22.22%	**100.00%**	**55.56%**	**100.00%**	22.22%
RVE	0.87%	**87.50%**	**26.92%**	77.78%	**26.92%**	33.33%	3.85%	**87.50%**	**26.92%**	**87.50%**	**26.92%**
VTHOG	2.24%	51.95%	59.70%	56.16%	61.19%	57.47%	**74.63%**	53.57%	67.16%	**60.76%**	71.64%
VTHOO	2.21%	52.38%	50.00%	63.64%	53.03%	63.08%	62.12%	**67.80%**	60.61%	66.67%	**66.67%**
VVHOG	1.44%	55.56%	58.14%	58.70%	62.79%	**65.85%**	62.79%	60.42%	**67.44%**	60.87%	65.12%
VVHOO	1.51%	57.89%	48.89%	62.86%	48.89%	**67.44%**	**64.44%**	60.98%	55.56%	62.16%	51.11%

Tabel 3.8 – Presisie en herroeping vir die etikette wat dikwels verwarring veroorsaak

Dit is interessant dat MXPOST vir die etiket VVHOO die beste telling vir presisie en herroeping het. Soos verwag, is TnT vir drie etikette gelyk aan of beter as Kombinasie2 vir beide presisie en herroeping.

3.4.5 RESULTATE VAN KOMBINASIE

Die resultate van die kombinering van die eerstevlakklassifiseerders MBT, SVM^{light}, MXPOST en TnT deur middel van die verskillende stemmetodes (afdeling 3.3) word in Tabel 3.9 gegee. Die akkuraatheid van die kombineerders (%), asook die foutkoersreduksie (Δ_{Err}) met betrekking tot die beste eerstevlakklassifiseerder (TnT) op UVVM en TEST word gegee.

Stemmetode	UVVM		TEST	
	%	Δ_{Err}	%	Δ_{Err}
TnT	80.80%	-	89.42%	-
Presisie-Herroeping – Opsie 1	81.08%	1.47%	89.12%	-2.85%
Presisie-Herroeping – Opsie 2	80.97%	0.88%	89.12%	-2.85%
Presisie-Herroeping – Opsie 3	80.97%	0.88%	89.22%	-1.90%
Presisie-Herroeping – Opsie 4	81.02%	1.17%	89.18%	-2.22%
Presisie-Herroeping – Opsie 5	80.97%	0.88%	89.25%	-1.58%
Etiket-Presisie – Opsie 6	80.57%	-1.17%	88.91%	-4.75%
Totale-Presisie – Opsie 7	81.02%	1.17%	89.32%	-0.95%

Tabel 3.9 – Resultate van die kombinasie van die eerstevlakklassifiseerders d.m.v. verskillende stemmetodes

Op die UVVM-datastel is daar vir alle metodes behalwe Etiket-Presisie – Opsie 6 ’n klein reduksie in die foutkoers terwyl die resultate op TEST ’n verhoging in die foutkoers toon. Tabel 3.12 en Tabel 3.13 in die volgende afdeling toon dat hierdie verskille nie statisties betekenisvol is nie, en hierdie resultate is nie verbasend na die bestudering van Tabel 3.4 op bladsy 51 en die bypassende bespreking nie. Tabel 3.10 sluit aan by Tabel 3.5 op bladsy 52 en die bypassende bespreking waar die eerstevlakklassifiseerder wat heelwat beter as die ander gevaar het (TnT) buite rekening gelaat word. Die foutkoersreduksie met betrekking tot die beste

eerstevlakklassifiseerder (MBT) wanneer ’n kombinasie van net die drie eerstevlakklassifiseerders MBT, SVM^{light} en MXPOST (voortaan Kombi 3) op beide toets-datastelle toegepas word, is aansienlik meer vir alle kombinasiemetodes met ’n minimum van 7.91% vir Totale-Presisie – Opsie 7 op UVVM en ’n maksimum van 18.48% vir Etiket-Presisie – Opsie 6 op TEST. Tabel 3.14 en Tabel 3.15 van die volgende afdeling toon dat hierdie verskille wel statisties betekenisvol is, en hierdie resultate stem dus ooreen met Tabel 3.5 op bladsy 52 en die bypassende bespreking.

Stemmetode	UVVM (Kombi 3)		TEST (Kombi 3)	
	%	Δ_{Err}	%	Δ_{Err}
MBT	77.93%	-	85.87%	-
Presisie-Herroeping – Opsie 1	81.14%	14.54%	88.38%	17.77%
Presisie-Herroeping – Opsie 2	81.14%	14.54%	88.38%	17.77%
Presisie-Herroeping – Opsie 3	81.02%	14.03%	88.41%	18.01%
Presisie-Herroeping – Opsie 4	81.08%	14.29%	88.41%	18.01%
Presisie-Herroeping – Opsie 5	81.14%	14.54%	88.38%	17.77%
Etiket-Presisie – Opsie 6	81.31%	15.31%	88.48%	18.48%
Totale-Presisie – Opsie 7	79.67%	7.91%	88.25%	16.82%

Tabel 3.10 – Resultate van die kombinasie van Kombi 3 d.m.v. verskillende stemmetodes

Let op dat Totale-Presisie – Opsie 7 heelwat swakker as die ander opsies vaar met die moeiliker taak (UVVM). Dit kan soos volg verklaar word.

Wanneer Totale-Presisie – Opsie 7 toegepas word, sal die etiket wat deur die beste eerstevlak-klassifiseerder (in hierdie geval MBT) toegeken is deurgaans in alle gevalle (vir alle etikette) deur die kombinasieklassifiseerder toegeken word, behalwe waar beide die minder akkurate eerstevlakklassifiseerders saamstem oor die etiket maar verskil van die beste eerstevlak-klassifiseerder. Wanneer geen van die drie klassifiseerders dus saamstem oor ’n etiket nie, sal die beste klassifiseerder se etiket altyd toegeken word. ’n Sekere mate van foutkoersreduksie vind plaas, aangesien die twee swakker klassifiseerders in sommige gevalle teen die beste klassifiseerder saamstaan en, in hierdie gevalle, die beste klassifiseerder meer gereeld korrigeer as nie. Hoe moeiliker die taak, hoe minder gereeld sal die swakker klassifiseerders saamstem om die beste klassifiseerder te korrigeer, aangesien ’n groter verspreiding van etikette deur die klassifiseerders toegeken sal word. Verder, hoe groter die verskil in totale presisie tussen die beste klassifiseerder en die swakker klassifiseerders, hoe minder gereeld korrigeer die swakker klassifiseerders die beste klassifiseerder in hierdie gevalle en hoe kleiner word die foutkoersreduksie wat plaasvind (soos gesien in die geval waar TnT ingesluit is).

Wanneer die ander opsies toegepas word, word die eerstevlakklassifiseerder wat die beste vaar met elke spesifieke etiket se klassifikasie deur die kombineerder gekies, behalwe wanneer beide die minder akkurate klassifiseerders (vir daardie spesifieke etiket) saamstem oor die etiket en verskil van die beste klassifiseerder. Daar is dus 'n groter mate van foutkoersreduksie aangesien 'n meer gesegmenteerde en dus meer ingeligte besluit gemaak word oor watter klassifiseerder die beste in verskillende gevalle is. Hoe groter die verskil in totale presisie tussen die beste klassifiseerder en die swakker klassifiseerders in hierdie geval, hoe meer van die segmente sal deur die beste klassifiseerder oorheers word en hoe minder variasie is daar ten opsigte van watter klassifiseerder vir watter etiket as die beste geag word, wat veroorsaak dat hierdie opsies meer ekwivalent aan die Totale-Presisie – Opsie 7 raak. Dit het tot gevolg dat die verskil in foutkoersreduksie tussen die opsies al kleiner word, weereens soos gesien in die geval waar TnT ingesluit is (verwys na Tabel 3.9 op bladsy 55).

Na aanleiding van die vorige resultate, soos in Tabel 3.10 weergegee, word Kombi 3 ook nou vergelyk met TnT as basislyn in Tabel 3.11 om te bepaal of die drie saam beter vaar as TnT, wat die akkuraatste eerstevlakklassifiseerder is. Die kombinasiemetodes toegepas op UVVM lewer in alle gevalle met die uitsondering van Totale-Presisie – Opsie 7 'n geringe foutkoersreduksie, met maksimum akkuraatheid vir Kombi 3 van 81.31% teenoor 80.80% vir TnT. Op die toetsdatastel TEST slaag Kombi 3 nie daarin om 'n foutkoersreduksie teweeg te bring nie, met maksimum akkuraatheid vir Kombi 3 van 88.48% teenoor 89.42% vir TnT.

Stemmetode	**UVVM (Kombi 3)**		**TEST (Kombi 3)**	
	%	Δ_{Err}	**%**	Δ_{Err}
TnT	80.80%	-	89.42%	-
Presisie-Herroeping – Opsie 1	81.14%	1.76%	88.38%	-9.81%
Presisie-Herroeping – Opsie 2	81.14%	1.76%	88.38%	-9.81%
Presisie-Herroeping – Opsie 3	81.02%	1.17%	88.41%	-9.49%
Presisie-Herroeping – Opsie 4	81.08%	1.47%	88.41%	-9.49%
Presisie-Herroeping – Opsie 5	81.14%	1.76%	88.38%	-9.81%
Etiket-Presisie – Opsie 6	81.31%	2.64%	88.48%	-8.86%
Totale-Presisie – Opsie 7	79.67%	-5.87%	88.25%	-11.08%

Tabel 3.11 – Die kombinasie van Kombi 3 in vergelyking met TnT

Die statistiese betekenisvolheid van hierdie resultate word vervolgens bepaal.

3.4.6 STATISTIESE BETEKENISVOLHEID

Die statistiese betekenisvolheid van die verskil tussen 'n eerstevlakklassifiseerder-basislyn en die verskeie kombinasiemetodes soos bespreek in die vorige afdeling word deur middel van McNemar se toets (afdeling 1.6.5) bereken. Die tabelle lys telkens die eerstevlakklassifiseerder-basislyn (Klassifiseerder A) met die kombinasieklassifiseerder waarvoor die statistiese betekenisvolheid bepaal moet word (Klassifiseerder B). Die aantal klassifikasievoorbeelde word telkens volgens die verwarringsmatriks in Tabel 1.2 op bladsy 10 getel ($\boldsymbol{n_{00}}, \boldsymbol{n_{01}}, \boldsymbol{n_{10}}, \boldsymbol{n_{11}}, \boldsymbol{n}$ notasies volgens (1.3) en (1.4) op bladsy 10) om die McNemar-statistiek se waarde te bereken, soos uiteengesit in vergelyking (1.5) op bladsy 10. Die McNemar-statistiek is benaderd χ^2 met 1 vryheidsgraad verdeel en 1 minus die waarde van die kumulatiewe digtheidsfunksie word dus gebruik om die waarskynlikheid dat die verskil tussen Klassifiseerder A en Klassifiseerder B se foutkoerse toevallig kan opduik indien hulle steekproewe uit populasies met gelyke parameters verteenwoordig (aangedui as $\boldsymbol{\rho}(\mathbf{A} = \mathbf{B})$) te beraam. Die foutkoersreduksie met betrekking tot die eerstevlakklassifiseerder (Δ_{Err}) soos in afdeling 3.4.5, wat in hierdie geval Klassifiseerder A is, word vir verwysingsdoeleindes in die onderstaande tabelle herhaal.

Klassifiseerder A (UVVM)	Klassifiseerder B (UVVM)	$\boldsymbol{n_{00}}$	$\boldsymbol{n_{10}}$	$\boldsymbol{n_{01}}$	$\boldsymbol{n_{11}}$	$\boldsymbol{n}$	McNemar statistiek	ρ(A=B)	Δ_{Err}
TnT	Presisie-Herroeping – Opsie 1	306	30	35	1405	1776	0.25	61.98%	1.47%
TnT	Presisie-Herroeping – Opsie 2	311	27	30	1408	1776	0.07	79.11%	0.88%
TnT	Presisie-Herroeping – Opsie 3	306	32	35	1403	1776	0.06	80.70%	0.88%
TnT	Presisie-Herroeping – Opsie 4	306	31	35	1404	1776	0.14	71.19%	1.17%
TnT	Presisie-Herroeping – Opsie 5	311	27	30	1408	1776	0.07	79.11%	0.88%
TnT	Etiket-Presisie – Opsie 6	311	34	30	1401	1776	0.14	70.77%	-1.17%
TnT	Totale-Presisie – Opsie 7	315	22	26	1413	1776	0.19	66.50%	1.17%

Tabel 3.12 – Die verskil tussen TnT en kombinasieklassifiseerders op UVVM

Klassifiseerder A (TEST)	Klassifiseerder B (TEST)	$\boldsymbol{n_{00}}$	$\boldsymbol{n_{10}}$	$\boldsymbol{n_{01}}$	$\boldsymbol{n_{11}}$	$\boldsymbol{n}$	McNemar statistiek	ρ(A=B)	Δ_{Err}
TnT	Presisie-Herroeping – Opsie 1	272	53	44	2617	2986	0.66	41.66%	-2.85%
TnT	Presisie-Herroeping – Opsie 2	268	57	48	2613	2986	0.61	43.50%	-2.85%
TnT	Presisie-Herroeping – Opsie 3	274	48	42	2622	2986	0.28	59.82%	-1.90%
TnT	Presisie-Herroeping – Opsie 4	274	49	42	2621	2986	0.40	52.94%	-2.22%
TnT	Presisie-Herroeping – Opsie 5	268	53	48	2617	2986	0.16	69.06%	-1.58%
TnT	Etiket-Presisie – Opsie 6	268	63	48	2607	2986	1.77	18.39%	-4.75%
TnT	Totale-Presisie – Opsie 7	284	35	32	2635	2986	0.06	80.70%	-0.95%

Tabel 3.13 – Die verskil tussen TnT en kombinasieklassifiseerders op TEST

Klassifiseerder A (UVVM)	Klassifiseerder B (UVVM)	n_{00}	n_{10}	n_{01}	n_{11}	n	McNemar statistiek	ρ(A=B)	Δ_{Err}
MBT	**Presisie-Herroeping – Opsie 1**	310	25	82	1359	1776	29.31	0.00%	14.54%
MBT	**Presisie-Herroeping – Opsie 2**	309	26	83	1358	1776	28.77	0.00%	14.54%
MBT	**Presisie-Herroeping – Opsie 3**	311	26	81	1358	1776	27.25	0.00%	14.03%
MBT	**Presisie-Herroeping – Opsie 4**	310	26	82	1358	1776	28.01	0.00%	14.29%
MBT	**Presisie-Herroeping – Opsie 5**	310	25	82	1359	1776	29.31	0.00%	14.54%
MBT	**Etiket-Presisie – Opsie 6**	309	23	83	1361	1776	32.84	0.00%	15.31%
MBT	**Totale-Presisie – Opsie 7**	348	13	44	1371	1776	15.79	0.01%	7.91%

Tabel 3.14 – Die verskil tussen MBT en Kombi 3 op UVVM

Klassifiseerder A (TEST)	Klassifiseerder B (TEST)	n_{00}	n_{10}	n_{01}	n_{11}	n	McNemar statistiek	ρ(A=B)	Δ_{Err}
MBT	**Presisie-Herroeping – Opsie 1**	314	33	108	2531	2986	38.84	0.00%	17.77%
MBT	**Presisie-Herroeping – Opsie 2**	314	33	108	2531	2986	38.84	0.00%	17.77%
MBT	**Presisie-Herroeping – Opsie 3**	316	30	106	2534	2986	41.36	0.00%	18.01%
MBT	**Presisie-Herroeping – Opsie 4**	316	30	106	2534	2986	41.36	0.00%	18.01%
MBT	**Presisie-Herroeping – Opsie 5**	316	31	106	2533	2986	39.97	0.00%	17.77%
MBT	**Etiket-Presisie – Opsie 6**	313	31	109	2533	2986	42.35	0.00%	18.48%
MBT	**Totale-Presisie – Opsie 7**	332	19	90	2545	2986	44.95	0.00%	16.82%

Tabel 3.15 – Die verskil tussen MBT en Kombi 3 op TEST

Tabel 3.12 en Tabel 3.13 sluit aan by Tabel 3.9 op bladsy 55 uit die vorige afdeling. TnT toon ’n laer foutkoers vergeleke met alle kombineerders op TEST, terwyl die kombineerders op UVVM ’n geringe foutkoersreduksie in vergelyking met TnT met die uitsondering van Opsie 6 toon. Waar 0% vir $\boldsymbol{\rho}(\mathbf{A}=\mathbf{B})$ sal dui op ’n statisties betekenisvolle verskil kan in Tabel 3.12 en Tabel 3.13 gesien word dat die kombinasieklassifiseerders selfs in die gevalle waar daar wel ’n foutkoersreduksie vir die kombinasieklassifiseerder is, geen statisties betekenisvolle verskil maak nie. Die laagste waarskynlikheid dat die foutkoersreduksie toevallig kan opduik indien die klassifiseerders steekproewe uit populasies met gelyke parameters verteenwoordig is vir Etiket-Presisie – Opsie 6 op TEST met 18.39%. Hierdie waarskynlikheid neem toe tot 80.70% op TEST en UVVM.

In aansluiting met Tabel 3.10 op bladsy 56 word die statistiese betekenisvolheid van Kombi 3 in Tabel 3.14 en Tabel 3.15 bepaal. Met MBT as basislyn is $\boldsymbol{\rho}(\mathbf{A}=\mathbf{B})$ in alle gevalle vir alle praktiese doeleindes 0%. Die foutkoerse van alle kombinasieklassifiseerders in Tabel 3.14 en Tabel 3.15 verskil dus statisties betekenisvol van dié van MBT en daar kan dus met groot vertroue gesê word dat die kombinering van eerstevlakklassifiseerders in hierdie geval ’n positiewe verskil gemaak het.

Laastens bepaal Tabel 3.16 en Tabel 3.17 die statistiese betekenisvolheid van die resultate in Tabel 3.11 op bladsy 57 waar Kombi 3 vergelyk word met TnT. Vir UVVM is daar geen statistiese betekenisvolle verskil tussen die foutkoerse van Klassifiseerder A en Klassifiseerder B nie, dus kan daar nie met sekerheid gesê word dat Klassifiseerder B beter vaar as Klassifiseerder A nie, alhoewel daar wel ’n foutkoersreduksie teenwoordig is vir al die opsies behalwe Totale-Presisie – Opsie 7. Vir TEST bestaan daar ’n negatiewe foutkoersreduksie vir al die opsies met betrekking tot Klassifiseerder A, en die verskil tussen die foutkoerse van Klassifiseerder A en Klassifiseerder B is deurgaans statisties betekenisvol. Daar kan dus met redelike sekerheid gesê word dat TnT beter vaar as Kombi 3 op TEST.

Klassifiseerder A (UVVM)	**Klassifiseerder B (UVVM)**	n_{00}	n_{10}	n_{01}	n_{11}	n	**McNemar statistiek**	ρ**(A=B)**	Δ_{Err}
TnT	**Presisie-Herroeping – Opsie 1**	268	67	73	1368	1776	0.18	67.26%	1.76%
TnT	**Presisie-Herroeping – Opsie 2**	267	68	74	1367	1776	0.18	67.48%	1.76%
TnT	**Presisie-Herroeping – Opsie 3**	269	68	72	1367	1776	0.06	79.98%	1.17%
TnT	**Presisie-Herroeping – Opsie 4**	269	67	72	1368	1776	0.12	73.44%	1.47%
TnT	**Presisie-Herroeping – Opsie 5**	267	68	74	1367	1776	0.18	67.48%	1.76%
TnT	**Etiket-Presisie – Opsie 6**	266	66	75	1369	1776	0.45	50.05%	2.64%
TnT	**Totale-Presisie – Opsie 7**	300	61	41	1374	1776	3.54	5.99%	-5.87%

Tabel 3.16 – Die verskil tussen TnT en Kombi 3 op UVVM

Klassifiseerder A (TEST)	**Klassifiseerder B (TEST)**	n_{00}	n_{10}	n_{01}	n_{11}	n	**McNemar statistiek**	ρ**(A=B)**	Δ_{Err}
TnT	**Presisie-Herroeping – Opsie 1**	246	101	70	2569	2986	5.26	2.18%	-9.81%
TnT	**Presisie-Herroeping – Opsie 2**	246	101	70	2569	2986	5.26	2.18%	-9.81%
TnT	**Presisie-Herroeping – Opsie 3**	246	100	70	2570	2986	4.95	2.61%	-9.49%
TnT	**Presisie-Herroeping – Opsie 4**	246	100	70	2570	2986	4.95	2.61%	-9.49%
TnT	**Presisie-Herroeping – Opsie 5**	247	100	69	2570	2986	5.33	2.10%	-9.81%
TnT	**Etiket-Presisie – Opsie 6**	247	97	69	2573	2986	4.39	3.61%	-8.86%
TnT	**Totale-Presisie – Opsie 7**	251	100	65	2570	2986	7.01	0.81%	-11.08%

Tabel 3.17 – Die verskil tussen TnT en Kombi 3 op TEST

3.4.7 DIE EFFEK VAN ’N VEREENVOUDIGDE ETIKETSTEL

Die laaste resultate wat bespreek word, is die effek wat ’n vereenvoudigde etiketstel op die akkuraatheid het. Die eerstevlakklassifiseerders en Kombinasie2 is in hierdie geval slegs volgens die hoofkategorieë op TEST geëvalueer. Let wel dat dieselfde klassifiseerders wat in die vorige evaluasies gebruik is ook hier gebruik word. Die klassifiseerders is dus met die volledige etiketstel afgerig en dit is net tydens die evaluasie waarin slegs hoofkategorieë in ag geneem word. Daar bestaan 13 hoofkategorieë naamlik <N> (naamwoord), <V> (werkwoord),

<A> (adjektief), <P> (voornaamwoord), <B> (bywoord), <T> (telwoord), <L> (lidwoord), <S> (setsel), <W> (tussenwerpsel), <K> (voegwoord), <R> (residu), <U> (uniek) en <Z> (punktuasie).

Akkuraatheid	**MBT**	**SVM*light***	**MXPOST**	**TnT**	**Kombinasie2**
Gemeet volgens volledige etiketstel	85.87%	82.38%	85.37%	89.42%	89.12%
Gemeet volgens hoofkategorieë	93.03%	88.78%	92.63%	95.45%	94.94%

Tabel 3.18 – Akkuraatheid gemeet volgens die volledige etiketstel en slegs die hoofkategorieë

Soos verwag, is daar ’n verbetering in die akkuraatheid vir alle klassifiseerders en bly die volgorde van prestasie onveranderd, naamlik van laag na hoog SVM*light*, MXPOST, MBT, Kombinasie2 en TnT.

3.5 SAMEVATTING

Hierdie hoofstuk begin deur die metode vir die implementering van ’n Afrikaanse kombinasie-woordsoortetiketteerder te verduidelik. Die kombinasie-woordsoortetiketteerder word saamgestel uit vier eerstevlakklassifiseerders wat elk in afdeling 3.2 bespreek word, naamlik

- MBT;
- SVM*light*;
- MXPOST; en
- TnT.

Geweegde stemming word gebruik om die eerstevlakklassifiseerders te kombineer. Afdeling 3.3 gee ’n verduideliking van die sewe opsies vir geweegde stemming wat verskil op grond van die manier waarop die gewigte aan die individuele eerstevlakklassifiseerders toegeken word. Die verskillende metodes vir die toekenning van die gewigte sluit in

- Totale-Presisie, wat die totale presisie as gewig toeken;
- Etiket-Presisie, wat die etiketpresisie van die voorgestelde etiket as gewig toeken; en
- Presisie-Herroeping, wat die voorgestelde etiket se etiketpresisie, asook die etiket-presisie van die etiket(te) wat voorgestel is deur die opposisie-eerstevlak-klassifiseerders in berekening bring wanneer gewigte toegeken word.

Afdeling 3.4 bespreek die resultate van die eerstevlakklassifiseerders individueel onder die volgende afdelings:

- Die kwaliteit van die eerstevlakklassifiseerders (afdeling 3.4.1) wat die akkuraatheid van elke eerstevlakklassifiseerder gee; en
- Ooreenkomste tussen eerstevlakklassifiseerders (afdeling 3.4.2) wat die patrone van ooreenkomste tussen die eerstevlakklassifiseerders bestudeer, aangesien kombinasie afhanklik is daarvan dat verskillende sisteme verskillende foute produseer.

Afdeling 3.4.4 bestudeer kombinasie in detail, terwyl daar na spesifieke tekseenhede en woordsoortetikette wat dikwels verwarring veroorsaak, gekyk word.

Die resultate van die samevoeging van die vier eerstevlakklassifiseerders word ook in afdeling 3.4 onder die volgende afdelings bespreek:

- Die teorie van kombinasie (afdeling 3.4.3) wat beraamde akkuraathede van toenemende vlakke van sukses bestudeer;
- Resultate van kombinasie (afdeling 3.4.5) wat die akkuraathede en foutkoers-reduksie van die kombinasie-woordsoortetiketteerder met betrekking tot ’n basislyn-eerstevlakklassifiseerder gee; en
- Statistiese betekenisvolheid (afdeling 3.4.6) wat die waarskynlikheid dat die verskil in foutkoerse van twee klassifiseerders toevallig kan opduik indien die klassifiseerders steekproewe uit populasies met gelyke parameters verteenwoordig, beraam. Vervolgens word afgelei dat die verskil net vir MBT as basislyn ’n statisties betekenisvolle verbetering teweeg bring - dit is vir Kombi 3 wanneer TnT nie in ag geneem word tydens kombinasie nie.

Laastens word die effek van ’n vereenvoudigde etiketstel in afdeling 3.4.7 voorgehou.

Hoofstuk 4

STRUKTURELE GEDEELTELIK-GEKONTROLEERDE LEER

Die gedeeltelik-gekontroleerde algoritme wat geïmplementeer word, is gebaseer op die strukturele leermetode, soos voorgestel deur Ando en Zhang (2005a) en Ando en Zhang (2005b). Die teoretiese agtergrond van die metode word eerstens bespreek, waarna ’n implementering en resultate weergegee word.

4.1 STRUKTURELE LEER

Strukturele leer verken die strukture wat daar in data bestaan deur dié strukture te ontgin wat oor die vermoë beskik om goeie voorspellings (Eng. *predictions*) te maak. In hierdie geval word dit spesifiek op die gedeeltelik-gekontroleerde probleem toegepas deurdat waardevolle inligting, of te wel strukture, uit ongeëtiketteerde data onttrek word. Die nuutgeleerde strukture beskik oor ’n goeie voorspellingsvermoë en word dan oorgedra na die teikenprobleem (Eng. *target problem*). Die teikenprobleem verwys na die gekontroleerde leerprobleem deur middel van geëtiketteerde data. In afdeling 2.1.1.1 word die probleem gedefinieer as ’n *klassifikasietaak*. Strukturele leer poog om ’n klassifiseerder te verbeter deur ooreenkomste (gedeelde strukture) tussen *meervoudige klassifikasietake* (Eng. *multiple prediction problems*) te vind en word ook dikwels na verwys as *multi-taak leer* (Eng. *multi-task learning*) (Caruana, 1997; Evgeniou & Pontil, 2007). Hierdie klassifikasietake word gegenereer uit ongeëtiketteerde data en die aanname word gemaak dat indien die strukture wat in ongeëtiketteerde data voorkom bruikbaar is en oor goeie voorspellingsvermoë beskik, dan sal dit ook bruikbaar wees vir geëtiketteerde data.

4.2 ’N MODEL VIR DIE LEER VAN STRUKTURE

Die algoritme wat aangewend word, pas ’n liniêre formulering van strukturele leer toe. Ter verduideliking word ’n standaard liniêre model voorgestel waarna dit uitgebrei word vir strukturele leer.

4.2.1 STANDAARD LINIÊRE MODEL

In die standaard formulering van gekontroleerde leer word gesoek na ’n klassifiseerder wat met invoervektor $\overline{x} \in X$ die ooreenkomstige uitvoer $y \in Y$ sal gee. Dit word voorgestel in (4.1) met klassifiseerder $f(\overline{x})$ waar $\overline{w}$ die *gewigvektor* is en $\overline{x}$ die *eienskapvektor* is.

$$f(\overline{x}) = \overline{w}^T\overline{x} \tag{4.1}$$

Afrigtingsdata wat bestaan uit ’n eindige versameling afrigtingsvoorbeelde $\{(\bar{X}_i, Y_i)\}$ word gewoonlik gebruik om ’n beramer $\hat{f}(\overline{x}) \in \mathrm{F}$ vir $f(\overline{x})$ te vind. ’n Algemene metode vir die akkurate beraming van $\hat{f}$ is *Empiriese Risiko Minimalisering* (ERM) (Eng. *Empirical Risk Minimization*) (Escudero *et al.*, 2000; Vapnik, 2000) wat ’n gewigvektor uit die afrigtingsdata aflei wat sodoende die voorspellingsfout (Eng. *prediction* error) op die geëtiketteerde afrigtingsdata sal minimeer.

’n Eienskapvektor word aan die hand van ’n voorbeeld geïllustreer. Gegee invoerdata “die kleurvolle blom” met $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ wat geïdentifiseer is as eienskappe. Die vektor vir die eienskap $Tekseenheid$ het die waarde 1 vir “kleurvolle”, terwyl alle ander bestaande tekseenhede, soos afgelei uit die afrigtingsdata, die waarde 0 het. Op dieselfde wyse het die vektor vir $Vorigetekseenheid$ ’n waarde 1 vir “die” en die vektor vir $Volgendetekseenheid$ ’n waarde 1 vir “blom”, terwyl alle ander tekseenhede ’n waarde 0 bevat. Die eienskapvektor $\overline{x}$ is ’n samevoeging van al drie die vermelde vektore:

$$\left.\begin{array}{r}
Tekseenheid = \text{“kleurvolle”} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \; \end{bmatrix} \\[2em]
Vorigetekseenheid = \text{“die”} \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ \; \end{bmatrix} \\[2em]
Volgendetekseenheid = \text{“blom”} \begin{bmatrix} \vdots \\ \vdots \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ \; \end{bmatrix}
\end{array}\right\} \bar{x}$$

Die hoë-dimensionele data het as gevolg van die hoë voorkoms van 0 teenoor 1 in die eienskapvektor ’n dataskaarsheidsprobleem tot gevolg, aangesien die meeste inskrywings bestaan uit 0 en daar dus nie veel inskrywings is om waar te neem nie. Die liniêre model vir strukturele leer is ’n uitbreiding op die standaard liniêre model wat spesifiek gebaseer is op strukturele leer en spreek die hoë-dimensionele probleem aan. Hierdie model word in die volgende afdeling bespreek.

4.2.2 LINIÊRE MODEL VIR STRUKTURELE LEER

Die *liniêre model vir strukturele leer* neem aan dat daar ’n lae-dimensionele voorspelling-struktuur wat deur die meervoudige klassifikasietake gedeel word, bestaan. Gestel daar bestaan m klassifikasietake wat aangedui word deur $l \in \{1, \ldots, m\}$. Die klassifiseerder vir klassifikasietaak l word soos volg voorgestel in (4.2).

$$f_l(\Theta, \bar{x}) = \bar{w}_l^T \bar{x} + \bar{v}_l^T \Theta \bar{x}, \qquad \Theta\Theta^T = \mathrm{I} \tag{4.2}$$

In die standaard liniêre model (afdeling 4.2.1) het elke klassifikasietaak slegs een komponent, $\overline{w}_l^T\overline{x}$ as invoer. Vir die liniêre model vir strukturele leer word egter 'n komponent, $\overline{v}_l^T\Theta\overline{x}$, wat taak-spesifiek is, bygevoeg, dus spesifiek tot klassifikasietaak l. I verteenwoordig die identiteits-matriks en die ortogonale matriks Θ is die gemeenskaplike strukturele parameter wat deur al die klassifikasietake gedeel word. $\overline{w}_l$ en $\overline{v}_l$ is gewigsvektore spesifiek vir elke klassifikasietaak l.

Die idee van die gesamentlike ERM-model is om 'n gemeenskaplike lae-dimensionele voorspellingstruktuur, wat geparametriseer is deur die projeksiematriks Θ, af te lei. Daar word gesoek na Θ en die gewigsvektore wat die som van die empiriese risiko oor al die klassifikasie-take sal minimeer. Θ is die lae-dimensionele projeksiematriks wat gedeel word deur alle klassifikasietake $l \in \{1, \dots, m\}$ en is 'n projeksie van 'n aanvanklike hoë-dimensionele ruimte na 'n lae-dimensionele ruimte met die doel om die afrigting van die uiteindelike klassifiseerder te vereenvoudig. Hierdie optimering word in (4.3) voorgestel, waar L die verliesfunksie verteenwoordig en die aantal eienskaptipes vir elke l aangedui word deur $i \in \{1, \dots, n_l\}$:

$$[\widehat{\Theta}, \{\hat{f}_l\}] = \underset{[\Theta,\{f_l\}]}{\arg\min} \sum_{l=1}^{m} \frac{1}{n_l} \sum_{i=1}^{n_l} L\left(f_l\left(\overline{x}_i^l\right), y_i^l\right). \tag{4.3}$$

4.2.3 WISSELSTRUKTUUROPTIMERINGSALGORITME

Die minimering van die som van die empiriese risiko oor al die klassifikasietake, soos vermeld in die liniêre model vir strukturele leer, word bereik deur die *wisselstruktuuroptimeringsalgoritme* (ASO) (Eng. *Alternating structure optimisation*). Die algoritme bestaan uit 'n twee-stap iteratiewe proses:

1. Rig m klassifiseerders onafhanklik af deur Θ vas te maak en die optimale klassifiseerders $(\overline{w}_l, \overline{v}_l)$ vir elke probleem l te vind; en
2. Vind die ooreenkomstige voorspellingstrukture in die m klassifiseerders deur die waardes van die gewigsvektore $\overline{w}_l$ en $\overline{v}_l$ vas te maak en Θ deur middel van *singulierewaarde-ontbinding* (SVD) (Eng. *singular value decomposition*) te bereken.

In die eerste stap is die klassifikasietake onafhanklik van mekaar – dit is eers in die tweede stap waar alle klassifikasietake gekoppel word deur middel van Θ. Die dimensiereduksie word bewerkstellig deur middel van SVD, en word uitgevoer op 'n stel voorspellers / klassifiseerders genaamd die voorspellingsruimte (Eng. *prediction space, a set of predictors*). Dit word moontlik gemaak deurdat daar meervoudige voorspellers vanuit meervoudige leertake (Eng. *learning tasks)* waargeneem word. As die waargenome voorspellers as steekproefpunte van die

voorspellerverdeling in die voorspellingsruimte (Eng. *sample points of the predictor distribution in the predictor space)* gesien word, kan die SVD-gebaseerde wisselstruktuur-optimeringsalgoritme (SVD-ASO) geïnterpreteer word as 'n algoritme wat die hoofkomponente of ooreenkomste van hierdie voorspellers vind, of anders gestel, uitvind "hoe goeie voorspellers lyk". Met ander woorde, die algoritme soek lae-dimensionele strukture wat oor die hoogste voorspellingsvermoë beskik.

Aangesien die doelwaarde van die optimeringsalgoritme met elke iterasie verminder, konvergeer die parameterwaardes na 'n lokale optimale oplossing.

4.3 EKSPERIMENTELE OPSTELLING

Vervolgens word geïllustreer hoe 'n algoritme vir die leer van strukture nuttig aangewend kan word vir gedeeltelik-gekontroleerde leer. Dit word gedoen deur meervoudige klassifikasietake met etikette vanuit ongeëtiketteerde data te skep. Hierdie klassifikasietake word *hulpprobleme* (Eng. *auxiliary problems*) genoem.

Die implementering is in GNU Octave (http://www.gnu.org/software/octave/) gedoen, terwyl Perl (http://www.perl.org/) hoofsaaklik vir die voorbereiding en onttrekking van data in 'n bruikbare formaat gebruik is.

Data

Die volgende afrigtingsdata is gebruik:

- 30000 korrek geëtiketteerde tekseenhede soos verskaf deur die Sentrum vir Tekstegnologie (CTexT®) van die Noordwes-Universiteit;
- 158450290 ongeëtiketteerde tekseenhede wat saamgestel is uit 17604090 tekseenhede afkomstig uit Die Burger en 140846200 tekseenhede uit Beeld, soos beskikbaar gestel deur die uitgewersmaatskappy Media24®. Die data is beskikbaar gestel in twee afsonderlike tekslêers van onderskeidelik 90,397kb en 742,197kb groot. Vanweë die grootte van die tekslêers was dit noodsaaklik om telkens 'n Perl-skrip te skryf vir die onttrekking van inligting asook die nodige teks-manipulering. In teenstelling met die geëtiketteerde afrigtingsdata bevat die rou, ongeëtiketteerde data heelwat ruis in die vorm van woorde wat nie deel uitmaak van die Afrikaanse taal nie, dus hoofsaaklik Engelse woorde, asook spel- en tikfoute, waarvan menige waarskynlik ontstaan het tydens die samevoeging van al die berigte in 'n enkele tekslêer per koerant, aangesien menige foute bestaan uit twee woorde wat vas geskryf is in plaas van los. Dit sal arbeidsintensief wees om

dié foute handmatig reg te stel en dit val ook buite die bestek van hierdie studie, aangesien daar eerder hoofsaaklik op die robuustheid van die algoritme staatgemaak word om die invloed van die ruis in die data te minimeer; en

- Die 30000 korrek geëtiketteerde tekseenhede is ewekansig verdeel in 80% afrigtingsdata, 10% toetsdata en 10% insteldata.

Eienskaponttrekking

Eienskapvektore word uit die hulpprobleme, asook die geëtiketteerde data gegenereer. Aangesien GNU Octave gebruik word vir die skep van die struktuurmatriks en die afrigting van die klassifiseerder is dit noodsaaklik dat die eienskapvektore numeries voorgestel word. Dit word bewerkstellig deur vooraf aan elke unieke tekseenheid en etiket ’n unieke indeks, wat in ’n leksikon gestoor word, toe te ken waaruit ’n eienskapvektor wat uit slegs numeriese waardes bestaan, geskep word. ’n Voorbeeld van eienskapvektore vir die generering van hulpprobleme word in Tabel 4.1 voorgehou. Die hoë voorkoms van 0 teenoor 1 in die eienskapvektor het ’n dataskaarsheidsprobleem tot gevolg, aangesien die meeste inskrywings bestaan uit 0 en daar dus nie veel inskrywings is om waar te neem nie. Hierdie probleem word juis aangespreek deur die generering van die struktuurmatriks. In hierdie studie word slegs die eienskappe $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ onttrek en in die eienskapvektor voorgestel. Ter wille van die praktiese uitvoerbaarheid van die eksperimente is slegs die tekseenhede wat in die geëtiketteerde data, sowel as die 1000 mees frekwente woorde, soos onttrek uit die ongeëtiketteerde afrigtingsdata in die leksikon wat gebruik is tydens die generering van die eienskapvektore ingesluit. Dit het tot gevolg dat ’n eienskapvektor in plaas van 3×1917627 rye net 3×5370 het.

Eienskapvektore		e1	e2	…	e400000
Tekseenheidindeks	**Vorigetekseenheid**				
	1	0	0	…	0
	.	.	.	…	.
	.	.	.	…	.
	.	.	1	…	.
	.	.	.	…	.
	.	1	.	…	.
	.	.	.	…	1
	⋮	⋮	⋮	…	⋮
	5370	0	0	…	0
	Volgendetekseenheid				
	1	0	0	…	0
	.	.	.	…	1
	.	1	.	…	.
	.	.	.	…	.
	.	.	.	…	.
	.	.	.	…	.
	.	.	1	…	.
	⋮	⋮	⋮	…	⋮
	5370	0	0	…	0
	Tekseenheid				
	1	0	0	…	0
	.	.	1	…	.
	.	.	.	…	.
	.	.	.	…	.
	.	.	.	…	1
	.	.	.	…	.
	.	1	.	…	.
	⋮	⋮	⋮	…	⋮
	5370	0	0	…	0

Tabel 4.1 – Voorstelling van eienskapvektore e1 tot e400000

Die prosedure

Die volgende gedeeltelik-gekontroleerde prosedure word gevolg in die eksperimentering.

1. Skep m hulpprobleme en ken hulpetikette aan **ongeëtiketteerde** data toe;
2. Bereken Θ (gedeelde struktuur) deur middel van gesamentlike ERM oor al die hulpprobleme; en
3. Rig die klassifiseerder af deur die waarde van Θ vas te maak en die empiriese risiko op die **geëtiketteerde** data vir die teikenprobleem te minimeer.

Vervolgens word elke stap van die prosedure in meer detail bespreek.

4.3.1 HULPPROBLEME

Hulpprobleme (Ando & Zhang, 2005a; Ando & Zhang, 2005b) met "geëtiketteerde afrigtingsdata" word outomaties vanuit ongeëtiketteerde data geskep.

Daar bestaan twee kriteria vir hulpprobleme:

1. Hulpprobleme moet soos die teikenprobleem lyk, met ander woorde daar moet ooreenkomste wees tussen die struktuur van die teikenprobleem en die struktuur van die hulpprobleem; en
2. Etikettering moet outomaties plaasvind. Die "antwoorde" moet beskikbaar wees sodat daar uit die struktuur geleer kan word.

Die hoofdoel van hulpprobleme is om 'n waarde vir die voorspellingstruktuur Θ vanuit ongeëtiketteerde data af te lei. Θ word gebruik in die uiteindelike teikenprobleem.

Hulpprobleme maak dit moontlik om bruikbare eienskapvektore, wat nie noodwendig in die geëtiketteerde data voorkom nie, vanuit ongeëtiketteerde data te ontdek. Alhoewel die hulpprobleme net van waarde kan wees, en 'n verhoging in die akkuraatheid van die uiteindelike klassifiseerder tot gevolg kan hê as daar ooreenkomste tussen die hulpprobleme en die teikenprobleem bestaan, is die metode egter wel robuus genoeg dat dit nie die akkuraatheid negatief sal beïnvloed as daar nie genoeg ooreenkomste tussen die hulpprobleme en die uiteindelike teikenprobleem bestaan nie.

Hulpprobleme word uit die ongeëtiketteerde data onttrek deur outomaties hulpetikette aan ongeëtiketteerde data toe te ken. Soos in Ando en Zhang (2005a) word net die 1000 mees frekwente tekseenhede wat in die ongeëtiketteerde data voorkom, gebruik vir die generering van die hulpprobleme (kyk Bylaag C). Vanuit die 1000 mees frekwente tekseenhede is aanvanklik 106851071 hulprye wat bestaan uit $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ onttrek. Die term *hulpry* word hier gebruik, omdat die ry nog nie op hierdie punt as hulpprobleem kwalifiseer nie, maar wel later in die proses aangewend word om

hulpprobleme te genereer. Hierdie groot hoeveelheid hulprye het die eksperimente nie prakties uitvoerbaar gemaak nie gegee die beskikbare rekenaarhulpbronne. Die aantal is toe verminder na 98825811 deur hulprye waarvan beide die $Vorigetekseenheid$ en $Volgendetekseenheid$ nie in die geëtiketteerde data voorkom nie uit te laat. Aangesien die aantal steeds heeltemal te veel was om die eksperimente prakties uitvoerbaar te maak is daar ewekansig slegs 400000 hulprye onttrek wat steeds 'n tydperk van ongeveer 48 uur vir die afrigting van die struktuurmatriks tot gevolg het.

Die pseudokode vir die generering van eienskapvektore waaruit hulpprobleme uiteindelik geskep word, word in Figuur 4.1 weergegee.

```
itereer deur alle ongeëtiketteerde tekseenhede
    stel teller = 1
    as huidige tekseenheid voorkom in 1000 frekwente tekseenhede
                               &&
    (vorige tekseenheid voorkom in leksikon
                               ||
     volgende tekseenheid voorkom in leksikon)

        stel Vorigetekseenheid{teller}  = vorige tekseenheid
        stel Volgendetekseenheid{teller}  = volgende tekseenheid
        stel Hulpetiket{teller} =  huidige tekseenheid
        vermeerder teller met 1
    einde as
    anders
        doen niks
einde iterasie
```

Figuur 4.1 – Onttrekking van hulprye

In hierdie studie word die ongekontroleerde sowel as die gedeeltelik-gekontroleerde strategie (Ando & Zhang, 2005a; Ando & Zhang, 2005b) vir die skep van hulpprobleme toegepas. Elke strategie word kortliks bespreek.

Ongekontroleerde strategie

Vir hierdie strategie word sommige substrukture van die ongeëtiketteerde data as hulpetikette gebruik, terwyl ander dele van die ongeëtiketteerde data gebruik word om hierdie etikette te klassifiseer. In hierdie studie word $Tekseenheid$ as hulpetiket aangewend, terwyl

$Vorigetekseenheid$ en $Volgendetekseenheid$ as eienskappe gegee word. Dit word aan die hand van ’n voorbeeld verder geïllustreer.

Gegee ’n eenvoudige ongeëtiketteerde sin: “Hy speel lekker.”

’n Aantal komplekse eienskapvektore word waargeneem, naamlik

$$\overline{a} = (\{Tekseenheid = \text{Hy}\}, \{Vorigetekseenheid =\}, \{Volgendetekseenheid = \text{ speel}\})$$

$$\overline{\text{b}} = (\{Tekseenheid = \text{speel}\}, \{Vorigetekseenheid = \text{Hy}\}, \{Volgendetekseenheid = \text{lekker}\})$$

$$\overline{\text{c}} = (\{Tekseenheid = \text{lekker}\}, \{Vorigetekseenheid = \text{speel}\}, \{Volgendetekseenheid = .\})$$

$$\overline{\text{d}} = (\{Tekseenheid = .\}, \{Vorigetekseenheid = \text{lekker}\}, \{Volgendetekseenheid = \})$$

Hulpprobleme word vir elke tekseenheidposisie geskep deur telkens $Tekseenheid$ as hulpetiket toe te ken en $Vorigetekseenheid$ en $Volgendetekseenheid$ as eienskappe te gebruik. Op hierdie manier word binêre klassifikasiegevalle uit ongeëtiketteerde data geskep deur sekere beskikbare eienskappe vir die doel van die hulpprobleem te verberg. Vir elke klassifikasiegeval wat bestaan uit $Vorigetekseenheid$, $Volgendetekseenheid$ word hulpprobleme soos voorgestel in Tabel 4.2 gegenereer wat aangewend word in die uiteindelike generering van die struktuur-matriks.

Is $Tekseenheid$ “”?
Is $Tekseenheid$ “Hy”?
Is $Tekseenheid$ “speel”?
Is $Tekseenheid$ “lekker”?
Is $Tekseenheid$ “.”?

Tabel 4.2 – Voorbeeld van die ongekontroleerde strategiehulpprobleme

Die ongekontroleerde strategie word in die volgende stappe opgesom:

1. Onttrek hulprye wat bestaan uit $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ uit die ongeëtiketteerde data;
2. Skep hulpprobleme vir die 400000 hulprye deur vir elke ry $Tekseenheid$ as hulp-etiket toe te ken terwyl $Vorigetekseenheid$ en $Volgendetekseenheid$ as eienskappe gebruik word;
3. Gebruik eienskappe $Vorigetekseenheid$ en $Volgendetekseenheid$ om Θ (gedeelde struktuur) vanuit die hulpprobleme te bereken; en

4. Rig die finale klassifiseerder met behulp van Θ en eienskappe $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ af.

Gedeeltelik-gekontroleerde strategie

Hierdie tegniek het sy oorsprong by die beginsel van deelleer (afdeling 2.1.3.2), maar die verskil is dat dit net in die generering van hulpprobleme gebruik word en nie tydens die afrigting van die finale klassifiseerder nie. Soos in Tabel 4.3 geïllustreer, word ’n werklike woordsoortetiket in plaas van ’n tekseenheid as hulpetiket toegeken.

Voorspel Klassifiseerder A “AOA” *vir hierdie klassifikasiegeval?*
Voorspel Klassifiseerder A “AOP” *vir hierdie klassifikasiegeval?*
Voorspel Klassifiseerder A “ASA” *vir hierdie klassifikasiegeval?*
.
.
.
Voorspel Klassifiseerder A “ZPR” *vir hierdie klassifikasiegeval?*

Tabel 4.3 – Voorbeeld van die gedeeltelik-gekontroleerde strategiehulpprobleme

Die strategie word in die volgende stappe opgesom:

1. Rig Klassifiseerder A met geëtiketteerde afrigtingsdata af. Gebruik net eienskap $Tekseenheid$ uit die afrigtingsdata;
2. Skep hulpprobleme vir die 400000 hulprye deur vir elke ry $Tekseenheid$ met Klassifiseerder A te etiketteer en die woordsoortetiket as hulpetiket toe te ken terwyl $Vorigetekseenheid$ en $Volgendetekseenheid$ as eienskappe gebruik word;
3. Gebruik eienskappe $Vorigetekseenheid$ en $Volgendetekseenheid$ om Θ (gedeelde struktuur) vanuit die hulpprobleme te bereken; en
4. Rig die finale klassifiseerder met behulp van Θ en eienskappe $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ af.

Die teorie in die sukses van die gebruik van hulpprobleme lê daarin dat eienskappe wat verborge eienskappe kan voorspel gekorreleerd met die verborge eienskappe is, en gevolglik ook gekorreleerd onder mekaar is. Hierdie tegniek help dus om gekorreleerde eienskappe wat oor ’n goeie voorspellingsvermoë beskik te identifiseer. Vir optimale resultate is dit natuurlik nodig om hulpetikette te kies wat ’n hoë korrelasie met die teikenprobleem het (Ando & Zhang, 2005a; Ando & Zhang, 2005b).

4.3.2 DIE GENERERING VAN DIE STRUKTUURMATRIKS

Die gedeelde struktuur Θ word deur middel van SVD soos bespreek in afdeling 4.2.3 bereken deur gebruik te maak van hulpprobleme wat uit ongeëtiketteerde data onttrek is. Dit behels gesamentlike ERM oor al die hulpprobleme. Die riglyne vir die implementering van die algoritme, soos weergegee in Ando en Zhang (2005a) en Ando en Zhang (2005b), dien as grondslag vir die implementering in hierdie studie.

4.3.2.1 PARAMETERS

h: Dimensie h is vasgestel op 50 en bepaal die hoeveelheid rye van die matriks Θ. Hoe meer rye Θ het, hoe minder reduksie van die probleem vind plaas. Met ander woorde hoe meer rye Θ het, hoe nader aan die hoë-dimensie oorspronklike probleem sal Θ wees. Die doel van Θ is om die oorspronklike hoë-dimensionele probleem te reduseer tot 'n lae-dimensionele probleem wat die belangrikste inligting behou en die inligting wat nie veel bydra tot die akkuraatheid nie uit te laat.

λ: Die reguleringsparameter λ is vasgestel op 0.0001. Die doel van λ is om te verhoed dat die model te kompleks word.

η: Die leertempo vir stogastiese gradiëntafdaling (SGD) (Eng. *stochastic gradient descent*) (Zhang, 2004) is vasgestel op 0.002.

4.3.2.2 UITVOER

Θ: Matriks Θ het h rye en dieselfde aantal kolomme as die lengte van die eienskapvektor.

Die dimensie van Θ is 50 x 10740.

Hoe meer rye Θ het, hoe nader aan die hoë-dimensie oorspronklike probleem sal dit wees. Die doel van Θ is om die oorspronklike hoë-dimensionele probleem te reduseer tot 'n lae-dimensionele probleem wat die belangrikste inligting behou en die inligting wat nie veel bydra tot die akkuraatheid van die klassifiseerder nie uit te laat.

4.3.2.3 ALGORITME

Die algoritme behels 'n twee-stap iteratiewe proses wat op die SVD-ASO-algoritme (afdeling 4.2.3) gebaseer is. In die eerste stap word m klassifiseerders onafhanklik van mekaar afgerig deur middel van SGD (Zhang, 2004). 'n SGD-algoritme evalueer die huidige datapunt en dateer die gewigsvektor ooreenkomstig op. SGD werk juis goed vir toepassings waar die hoeveelheid

datapunte en die dimensie van die probleem beide groot is. Die optimale voorspellingsresultaat kan bereik word deur ’n klein aantal iterasies oor die afrigtingsdata (Zhang, 2004). Die leertempo η is vasgestel op ’n vooraf bepaalde klein konstante en die SGD-metode word herhaal totdat konvergensie bereik is. Zhang (2004) bewys dat SGD verkieslik net vir ’n klein hoeveelheid iterasies herhaal moet word, daarom kan SGD in die praktyk effektief as afrigtingsmetode op grootskaalse voorspellingsprobleme gebruik word. Die verliesfunksie wat deur die SGD-metode gebruik word, is ’n aangepaste Huber-verliesfunksie (4.1) wat spesifiek bedoel is vir binêre klassifikasieprobleme waar $y \in \{\pm 1\}$ (Zhang, 2004).

$$L(p, y) = \begin{cases} max(0,1 - py)^2 \text{ as } py \geq -1 \\ -4py \text{ anders} \end{cases} \tag{4.1}$$

Die SGD-metode beweeg in elke stap in die rigting van steilste afdaling. Om hierdie rede word die afgeleide van $L(p, y)$ na p benodig, wat gegee word deur (4.2).

$$\frac{dL(p, y)}{dp} = \begin{cases} 0 \text{ as } py \geq 1 \\ -4y \text{ as } py < -1 \\ -2y(1 - py) \text{ anders} \end{cases} \tag{4.2}$$

In die tweede stap word die ooreenkomstige voorspellingstrukture in die m klassifiseerders gevind deur middel van SVD. SVD onttrek die relevante inligting en sluit die h mees relevante vektore in Θ in deur die rye van Θ gelyk te stel aan die h links singuliere vektore van U soos afgelei deur middel van SVD waar $U = [u_1, u_2, \ldots, u_m]$ met $\bar{u}_l = \hat{w}_l + \Theta^{\mathrm{T}}\bar{v}_l$. Dit is gelykstaande aan die h grootste singuliere waardes. SVD word op elke eienskaptipe onderskeidelik toegepas, maar die aantal eienskaptipes is vir die doel van hierdie studie vasgestel op 1, naamlik die tekseenheidposisie $Vorigetekseenheid$ en $Volgendetekseenheid$. Soos in afdeling 4.2.3 bespreek is, konvergeer die parameterwaardes $\{\hat{w}_l, \hat{v}_l\}, \Theta$ na ’n lokale optimale oplossing.

Die pseudokode vir die generering van die struktuurmatriks deur middel van die toepassing van SGD en SVD word in Figuur 4.2 weergegee. Die kode word nie geïtereer nie, omdat volgens Ando en Zhang (2005a) dit in die praktyk voldoende is om na die eerste iterasie by die waarde van Θ te volstaan, aangesien die dimensiereduksie voldoende deur middel van SVD gedurende die eerste iterasie plaasvind.

Die unieke etikette verskil vir die onderskeie strategieë, soos voorgestel in afdeling 4.3.1. Vir die ongekontroleerde strategie bestaan die unieke etikette uit die 1000 mees frekwente woorde (kyk Bylaag C), terwyl vir die gedeeltelik-gekontroleerde strategie die etikette die werklike stel

woordsoortetikette (kyk Bylaag B) soos wat vir die uiteindelike klassifiseerder gebruik word, is. Daar word vir elke strategie ’n afsonderlike struktuurmatriks gegenereer.

```
onttrek eienskapvektore uit alle hulprye

itereer deur alle eienskaptipes
     stel U = 0
     stel W = 0
     genereer lukraak waardes vir Theta
     itereer deur die unieke etikette
          stel v = Theta * u{etiket}
          itereer deur alle eienskapvektore
               as eienskapvektor{hulpetiket} gelyk is aan etiket
                    stel y = 1
               anders
                    stel y = -1

               w{etiket} = w{etiket} –leertempo * (2 *
               reguleringsparameter * w{etiket} + dLdp(w{etiket}'
               * eienskapvektor + v * Theta * eienskapvektor, y)
               * eienskapvektor)
          einde iterasie
          stel u{etiket} = w{etiket} + Theta' * v
     einde iterasie
     stel die rye van Theta = h links singuliere vektore van U
einde iterasie
```

Figuur 4.2 – Pseudokode vir generering van die struktuurmatriks

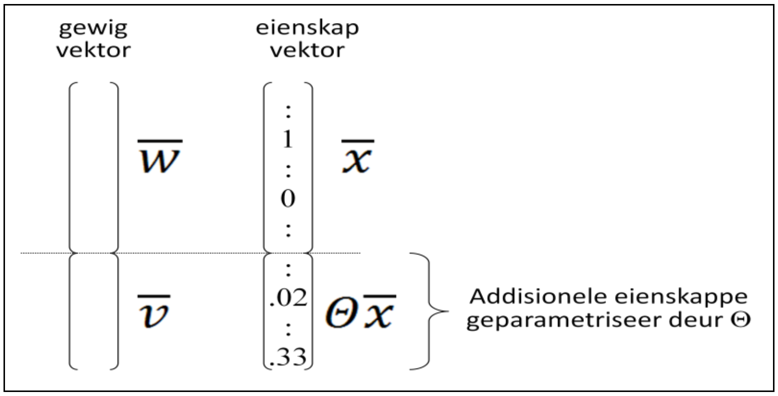
4.3.3 DIE AFRIGTING VAN ’N KLASSIFISEERDER

Tydens die afrigting van die klassifiseerder verskuif die fokus vanaf die ongeëtiketteerde data na die geëtiketteerde data, terwyl Θ die bruikbare inligting wat uit die ongeëtiketteerde data onttrek is verteenwoordig. ’n Voorspeller vir die teikenprobleem, naamlik die toekenning van ’n woordsoortetiket, word afgerig deur ERM op die oorspronklike geëtiketteerde data toe te pas. Θ bly nou vas, terwyl optimering met betrekking tot gewigte $\overline{w}$ en $\overline{v}$ plaasvind.

Die voorspeller, soos voorgestel in (4.3), is ’n standaard liniêre klassifikasieprobleem waarby addisionele eienskapvektore by $\Theta\overline{x}$ gevoeg is.

$$f(\Theta,\overline{x}) = \overline{w}^T\overline{x} + \overline{v}^T\Theta\overline{x} \tag{4.3}$$

Let op dat (4.3) ekwiwalent is aan die voorspeller in (4.1) op bladsy 64 waar $\overline{w}$ en $\overline{x}$ vervang is met $\left[\frac{\overline{w}}{\overline{v}}\right]$ en $\left[\frac{\overline{x}}{\Theta\overline{x}}\right]$, soos grafies voorgestel in Figuur 4.3.



Figuur 4.3 – Gewig- en eienskapvektore

Die pseudokode vir die afrigting van die voorspeller word in Figuur 4.4 uiteengesit.

```
onttrek eienskapvektore uit alle klassifikasiegevalle uit die
geëtiketteerde data

itereer deur alle eienskaptipes
     stel W = 0
     stel V = 0
     itereer deur die unieke etikette
          itereer deur alle eienskapvektore
               as eienskapvektor{hulpetiket} gelyk is aan etiket
                    stel y = 1
               anders
                    stel y = -1

               stel w_v{etiket} = voeg gewigsvektor_w{etiket} en
               gewigsvektor_v{etiket} saam
               stel Thetax = Theta * eienskapvektor
               stel x_theta_x = voeg eienskapvektor en Thetax'
               saam
               w_v{etiket} = w_v{etiket} - leertempo *(2 *
               reguleringsparameter * w_v{etiket} +
               dLdp(w_v{etiket}' * x_theta_x,y) * x_theta_x)
          einde iterasie
     einde iterasie
einde iterasie
```

Figuur 4.4 – Pseudokode vir die afrigting van ’n voorspeller

4.3.4 DIE KLASSIFISEERDER

Vir elke tekseenheid waaraan ’n etiket toegeken moet word, word daar deur al die moontlike etikette itereer sodat die voorspeller (4.3) op bladsy 77 telkens ’n waarde vir ‘n etiket bepaal. Die etiket met die hoogste waarde word uiteindelik aan die tekseenheid ter sprake toegeken. Die pseudokode van die proses word in Figuur 4.5 voorgelê.

```
onttrek eienskapvektore uit alle tekseenhede wat geëtiketteer moet
word

itereer deur alle eienskapvektore
      stel maks = -10000
      itereer deur die unieke etikette
            stel Thetax = Theta * eienskapvektor
            stel x_theta_x = voeg eienskapvektor en Thetax' saam
            fx = w_v{etiket}' * x_theta_x

            as fx groter is as maks
                  stel maks = fx
                  stel eienskapvektor{etiket} = etiket
            einde as
      einde iterasie
einde iterasie
```

Figuur 4.5 – Pseudokode vir klassifiseerder

4.3.5 RESULTATE

Die resultate van woordsoortetikettering deur middel van gedeeltelik-gekontroleerde leer word vervolgens bespreek. Die SVD-ASO-algoritme is eerstens net op geëtiketteerde data toegepas om die gekontroleerdeleerbasislyn te vorm waaruit die foutkoersreduksie vir die verskeie gedeeltelik-gekontroleerde leervariasies bereken word. Die variasies behels ’n afwisseling tussen die tipe hulpprobleme wat gebruik word en die tipe eienskappe wat in ag geneem word. Hulpprobleme is deur middel van die ongekontroleerde strategie (**Aux1**) en die gedeeltelik-gekontroleerde strategie (**Aux2**) uit die ongeëtiketteerde data onttrek waarna ’n struktuurmatriks Θ vir elkeen van die hulpprobleemstrategieë bereken is. Vir die afrigting van die gekontroleerde leerklassifiseerder word die struktuurmatriks $\Theta = 0$ gestel. Vir die afrigting van die gedeeltelik-

gekontroleerde leer **Aux1 + Aux2** klassifiseerder word daar twee struktuurmatrikse gebruik. Die eienskapvektore wat in ag geneem word, word ook afgewissel tussen eienskapvektore $Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$ (**x1**), $Vorigetekseenheid$ en $Volgendetekseenheid$ (**x2**) en net $Tekseenheid$ (**x3**). Laasgenoemde **x3** word net gebruik vir die afrigting van ’n gekontroleerde leerklassifiseerder tydens die generering van die **Aux2** hulpprobleme. Let wel dat vir die afrigting en uiteindelike klassifikasie waarin **x3** van toepassing is, word steeds net $Vorigetekseenheid$ en $Volgendetekseenheid$ saam met Θ gebruik, aangesien net hierdie eienskappe in die generering van Θ gebruik is. $Tekseenheid$ word net as eienskap in die gekontroleerde deel van die afrigting gebruik. Die akkuraatheid van die verskillende klassifiseerders (%) en foutkoersreduksie met betrekking tot die gekontroleerde leerbasislyn (Δ_{Err}) word in Tabel 4.4 weergegee.

SVD-ASO Resultate	**x1**		**x2**		**x3**	
	%	Δ_{Err}	**%**	Δ_{Err}	**%**	Δ_{Err}
Gekontroleerde leer	82.77%	-	37.80%	-	78.05%	-
Gedeeltelik-gekontroleerde leer Aux1	82.91%	0.78%	38.84%	1.67%	-	-
Gedeeltelik-gekontroleerde leer Aux2	82.98%	1.17%	-	-	-	-
Gedeeltelik-gekontroleerde leer Aux1 + Aux2	82.84%	0.39%	-	-	-	-

Tabel 4.4 – Woordsoortetikettering akkuraatheid en foutkoersreduksie

Alle variasies van die gedeeltelik-gekontroleerde leerklassifiseerder toon ’n foutkoersreduksie met betrekking tot die gekontroleerde leerbasislyn. Die klassifiseerder met **x2** eienskapvektore en **Aux1** wat hulpprobleme volgens die ongekontroleerde strategie genereer, het die hoogste foutkoersreduksie van 1.67%. Tabel 4.5 in die volgende afdeling toon dat hierdie die enigste statisties betekenisvolle foutkoersreduskie is. Vir klassifiseerders wat **x1** eienskapvektore gebruik, lewer **Aux2** wat hulpprobleme volgens die gedeeltelik-gekontroleerde strategie genereer die grootste foutkoersreduksie, naamlik 1.17%. Die kombinering van hulpprobleme met **Aux1** en **Aux2** het wel ’n foutkoersreduksie van 0.39%, maar dis nie, soos wat moontlik verwag sou word, ’n groter foutkoersreduksie as waar **Aux1** en **Aux2** apart gebruik word nie. Die volgende afdeling bespreek die statistiese betekenisvolheid van hierdie foutkoersreduksies.

4.3.6 STATISTIESE BETEKENISVOLHEID

Die statistiese betekenisvolheid van die verskil tussen die gekontroleerde leerbasislyn en die verskeie gedeeltelik-gekontroleerde leervariasies, soos bespreek in die vorige afdeling word deur middel van McNemar se toets (afdeling 1.6.5) bereken. Tabel 4.5 lys telkens die gekontro-

leerde leerbasislynklassifiseerder (Klassifiseerder A) met die gedeeltelik-gekontroleerde klassifiseerder waarvoor die statistiese betekenisvolheid bepaal moet word (Klassifiseerder B). Die aantal klassifikasievoorbeelde word telkens volgens die verwarringsmatriks in Tabel 1.2 op bladsy 10 getel ($\boldsymbol{n_{00}}, \boldsymbol{n_{01}}, \boldsymbol{n_{10}}, \boldsymbol{n_{11}}, \boldsymbol{n}$ notasies volgens (1.3) en (1.4) op bladsy 10) om die McNemar-statistiek se waarde te bereken soos uiteengesit in vergelyking (1.5) op bladsy 10. Die McNemar-statistiek is benaderd χ^2 met 1 vryheidsgraad verdeel en 1 minus die waarde van die kumulatiewe digtheidsfunksie word dus gebruik om die waarskynlikheid dat die verskil tussen Klassifiseerder A en Klassifiseerder B se foutkoerse toevallig kan opduik indien hulle steekproewe uit populasies met gelyke parameters verteenwoordig (aangedui as $\boldsymbol{\rho}(\mathbf{A} = \mathbf{B})$) te beraam.

Klassifiseerder A (gekontroleerde leer)	**Klassifiseerder B (gedeeltelik-gekontroleerde leer)**	n_{00}	n_{10}	n_{01}	n_{11}	n	**McNemar statistiek**	ρ**(A=B)**
x1	**Aux1 + x1**	507	3	7	2467	2984	0.90	34.28%
x2	**Aux1 + x2**	1822	3	34	1125	2984	24.32	0.00%
x1	**Aux2 + x1**	506	2	8	2468	2984	2.50	11.38%
x1	**Aux1/Aux2 +x1**	504	8	10	2462	2984	0.06	81.37%

Tabel 4.5 – Die verskil tussen gekontroleerde en gedeeltelik-gekontroleerde leer

Die waarskynlikheid dat die foutkoersreduksie toevallig kan opduik indien Klassifiseerder A en Klassifiseerder B steekproewe uit populasies met gelyke parameters verteenwoordig is vir **Aux1 + x2** vir alle praktiese doeleindes 0%, daarom is die verskil tussen die twee klassifiseerders statisties betekenisvol. Daar kan dus met groot vertroue gesê word dat die hulpprobleme wat uit die ongeëtiketteerde data genereer is volgens die ongekontroleerde strategie vir hierdie geval beslis ’n positiewe verskil gemaak het.

Vir die ander variasies van die gedeeltelik-gekontroleerde klassifiseerders is die verskil nie statisties betekenisvol nie. Vir **Aux2 + x1** is die waarskynlikheid dat die foutkoersreduksie toevallig kan opduik indien die klassifiseerders steekproewe uit populasies met gelyke parameters verteenwoordig 11.38%. Hierdie waarskynlikheid neem toe tot 34.28% vir **Aux1 + x1** en 81.37% vir **Aux1/Aux2 + x2**. Die beduidende verskil vir **Aux1 + x2** kan moontlik soos volg verklaar word. Uit die resultate van Klassifiseerder A in Tabel 4.4 word afgelei dat daar heelwat minder nuttige inligting beskikbaar is wanneer Klassifiseerder A met **x2** ($Vorigetekseenheid$ en $Volgendetekseenheid$) afgerig is, as wanneer Klassifiseerder A met **x3** ($Tekseenheid$) of **x1** ($Tekseenheid$, $Vorigetekseenheid$ en $Volgendetekseenheid$) as eienskapvektor afgerig word, met resultate van 37.80% teenoor 78.05% of 82.77%. Dit is duidelik dat $Tekseenheid$ beduidend

meer nuttige inligting bevat as $Vorigetekseenheid$ en $Volgendetekseenheid$. Gevolglik verskaf die byvoeging van die struktuurmatriks, wat vir alle gevalle afgerig is met die beskikbaarheid van die beperkte inligting vervat in eienskapvektor **x2**, nie genoegsame addisionele inligting om ’n beduidende verskil te maak in die geval waar **x1** reeds beskikbaar is nie, maar wel genoegsame addisionele inligting om ’n beduidende verskil te maak in die geval waar slegs die beperkte inligting vervat in **x2** beskikbaar is.

Verdere redes vir die resultate en metodes om te verbeter word as deel van die vooruitskouing in Hoofstuk 5 bespreek.

4.4 SAMEVATTING

In hierdie hoofstuk is strukturele leer aan die hand van ’n model vir die leer van strukture voorgestel. Hierdie model word verduidelik deur eerstens in afdeling 4.2.1 ’n standaard liniêre model voor te stel en dan hierdie model uit te brei in afdeling 4.2.2 na ’n model vir strukturele leer.

Afdeling 4.2.3 stel ’n ASO-algoritme voor wat strukturele leer op die probleem van gedeeltelik-gekontroleerde leer vir klassifikasie toepas. Dimensiereduksie vind plaas deur middel van SVD wat die mees ooreenkomstige voorspellingstrukture vind.

Die eksperimentele opstelling in afdeling 4.3 stel die data, eienskapvektore en prosedure wat gevolg word vir die implementering van die SVD-ASO-algoritme voor. Die prosedure word soos volg opgesom:

- Die onttrekking van hulpprobleme uit ongeëtiketteerde data deur middel van ’n ongekontroleerde of gedeeltelik-gekontroleerde strategie (afdeling 4.3.1);
- Die generering van ’n struktuurmatriks uit die hulpprobleme (afdeling 4.3.2); en
- Die afrigting van ’n klassifiseerder op geëtiketteerde data en die struktuurmatriks (afdeling 4.3.3).

In afdeling 4.3.4 word die implementering en pseudokode vir die gedeeltelik-gekontroleerde klassifiseerder voorgehou.

Die hoofstuk sluit af deur die akkuraathede en foutkoersreduksie met betrekking tot die gekontroleerde leerbasislyn in afdeling 4.3.5 voor te hou waarna die statistiese betekenisvolheid van hierdie resultate in afdeling 4.3.6 ondersoek word.

Hoofstuk 5

SLOT

5.1 SAMEVATTING

Die oorhoofse doel van hierdie studie was om te bepaal of dit moontlik is om vir ’n gegewe taal wat beskik oor beperkte hulpbronne die akkuraatheid van ’n klassifiseerder te verhoog sonder om duur, arbeidsintensiewe hulpbronne, soos geëtiketteerde data, by te voeg. Afrikaans dien as hulpbronskaars taal en woordsoortetikettering is gekies as klassifikasietaak. Daar is met twee verskillende metodes geëksperimenteer om die doel te bereik, naamlik

- Die kombinasie van klassifiseerders; en
- Die toepassing van ’n gedeeltelik-gekontroleerde strukturele leermetode wat gebruik maak van geëtiketteerde, sowel as ongeëtiketteerde data.

Hoofstuk 1 begin deur die ontwikkeling van taaltegnologiese toepassings met behulp van hulpbronne voor te stel. Afrikaans en die ander nege inheemse tale van Suid-Afrika word as hulpbronskaarstale geïdentifiseer. Die volgende doelstellings, wat as onderbou vir die daaropvolgende hoofstukke dien, word gestel, naamlik

1. Om te bepaal of kombinasietegnieke effektief gebruik kan word om die akkuraatheid van ’n Afrikaanse woordsoortetiketteerder te verbeter, word vier woordsoortetiketteerders wat deur middel van verskillende masjienleeralgoritmes ontwikkel is in ’n kombinasie-woordsoortetiketteerder saamgevoeg; en
2. Om te bepaal of gedeeltelik-gekontroleerde leer effektief is, word ’n algoritme geïmplementeer wat op gedeeltelik-gekontroleerde leer gebaseer is en word dit gebruik om ’n kerntegnologie, in hierdie geval ’n woordsoortetiketteerder vir Afrikaans, af te rig.

Verder stel Hoofstuk 1 ook ’n reeks evaluasiemetrieke voor wat van toepassing is in die studie en ook algemeen vir die probleem van woordsoortetikettering gebruik word. Hierdie evaluasiemetrieke is presisie, herroeping, akkuraatheid, foutkoers, foutkoersreduksie en McNemar se toets vir die bepaling van statistiese betekenisvolheid.

Hoofstuk 2 bespreek gekontroleerde, ongekontroleerde en gedeeltelik-gekontroleerde leer binne die konteks van masjienleer. Ongekontroleerde leer waarvoor voorbeelde van volledige klassifikasiegevalle nie verskaf word nie, en etikette dus uit die afrigtingsdata afgelei moet word, word oorsigtelik bespreek.

Vir gekontroleerde leer word klassifikasie as fokus geïdentifiseer, aangesien woordsoortetikettering ’n klassifikasietaak is wat uit ’n klein versameling diskrete waardes (woordsoort-

etikette) ’n klassifikasie toeken. Twee klassifikasie-algoritmes wat algemeen in die literatuur voorkom, is die naïewe Bayes-algoritme en die *k*-naastebuurpuntalgoritme.

Gedeeltelik-gekontroleerde leer verwys na leerprobleme waarvoor daar ’n klein hoeveelheid geëtiketteerde afrigtingsdata en ’n groot hoeveelheid ongeëtiketteerde afrigtingsdata beskikbaar is. Bekende gedeeltelik-gekontroleerde algoritmes sluit in selfleer, deelleer, die aanwending van generatiewe modelle, S3VM en grafiekgebaseerde metodes.

In die tweede deel van Hoofstuk 2 word verskeie metodes vir die kombinering van klassifikasie-algoritmes uiteengesit. Dit sluit in die kaskademetode, die stemmetode wat eenvoudige of geweegde stemming kan wees, versterking, groepering, opstapeling, klassifiseerderomruiling, ’n volgordesisteem en interne kombinasie.

Hoofstuk 3 beskryf vier woordsoortetiketteerders, naamlik MBT, SVM*light*, MXPOST en TnT, wat saamgevoeg word deur middel van geweegde stemming om ’n Afrikaanse kombinasie-woordsoortetiketteerder te vorm. Die kombinasie-woordsoortetiketteerder bied sewe opsies wat verskil op grond van die manier waarop die gewigte aan die individuele eerstevlak-klassifiseerders toegeken word. Die gewigstoekenning wissel tussen Totale-Presisie, wat die totale presisie as gewig toeken, Etiket-Presisie, wat die etiketpresisie van die voorgestelde etiket as gewig toeken, en Presisie-Herroeping wat die voorgestelde etiket se etiketpresisie, asook die etiketpresisie van die etiket(te) wat voorgestel is deur die opposisie-eerstevlak-klassifiseerders in berekening bring wanneer gewigte toegeken word.

Die resultate van die eerstevlakklassifiseerders, sowel as die kombinasie-woordsoort-etiketteerder behels die bestudering van die uitvoer op verskeie vlakke.

Vir die eerstevlakklassifiseerders lewer TnT die beste resultate (akkuraatheid 80.80% en 89.42%), terwyl die beste kombinasie-etiketteerder net in sommige gevalle beter as TnT vaar (foutkoersreduksie 1.47%). Aangesien die resultate daarop dui dat kombinasie nie ’n statisties betekenisvolle verbetering op TnT tot gevolg het nie word TnT vir ’n paar kombinasie-eksperimente buite rekening gelaat. Dit bring wel ’n statisties betekenisvolle verbetering met betrekking tot MBT (foutkoersreduksie 18.48%) teweeg.

In Hoofstuk 4 word strukturele leer binne die konteks van gedeeltelik-gekontroleerde leer toegepas deur die implementering van die SVD-ASO-algoritme. Hulpprobleme wat lyk soos die teikenprobleem word uit die ongeëttiketteerde data deur middel van ’n gekontroleerde of ongekontroleerde strategie onttrek. Die liniêre model vir strukturele leer neem aan dat daar ’n lae-dimensionele voorspellingstruktuur, wat deur die meervoudige klassifikasietake gedeel word, bestaan. Hierdie lae-dimensionele struktuurmatriks wat gedeel word deur alle klassifikasietake – in hierdie geval die hulpprobleme – is ’n projeksie van ’n aanvanklike hoë-

dimensionele ruimte na ’n lae-dimensionele ruimte met die doel om die afrigting van die uiteindelike woordsoortetiketteerder op geëtiketteerde data te vereenvoudig.

In die eksperimente word ’n gekontroleerde leer woordsoortetiketteerder met ’n struktuurmatriks van 0 as basislyn vir die resultate afgerig. Dus word geen hulpprobleme uit die ongeëtiketteerde data vir die basislyn in ag geneem nie.

Die byvoeging van hulpprobleme uit die ongeëtiketteerde data toon net vir **Aux1 + x2** ’n statisties betekenisvolle verskil (foutkoersreduksie 1.67%).

5.2 GEVOLGTREKKING

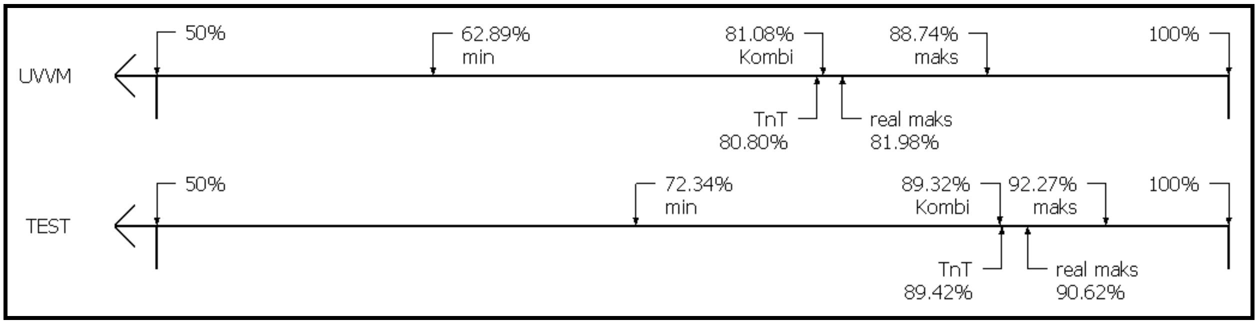
Die navorsingsvrae soos gestel in Hoofstuk 1 word weer voorgehou en kortliks beantwoord:

1. Kan kombinasietegnieke gebruik word om die akkuraatheid van ’n Afrikaanse woordsoortetiketteerder te verbeter?

Uit die resultate van afdelings 3.4.5 en 3.4.6 word afgelei dat die samevoeging van eerstevlak-klassifiseerders in ’n kombinasie-woordsoortetiketteerder net in sekere gevalle ’n verbetering toon. Alhoewel die kombinasie-woordsoortetiketteerder ’n geringe foutkoersreduksie op TnT vir UVVM getoon het (foutkoersreduksie 1.47%), kon kombinasietegnieke nie suksesvol in hierdie studie aangewend word om ’n statisties betekenisvolle verbetering op die beste eerstevlak-klassifiseerder, naamlik TnT te behaal nie. Uit verdere eksperimente kan wel afgelei word dat kombinasie tog nuttig blyk te wees wanneer die eerstevlakklassifiseerders se akkuraathede nader aan mekaar is (verwys na Tabel 3.2 op bladsy 50 en Tabel 3.10 op bladsy 56). Hierdie kombinasie van die drie swakker eerstevlakklassifiseerders (Kombi 3) vaar beter as enige van die eerstevlakklassifiseerders waaruit dit opgebou is, maar slaag nie daarin om vir UVVM ’n statisties betekenisvolle verbetering op die beste eerstevlakklassifiseerder (TnT) te bereik nie (verwys na Tabel 3.16 en Tabel 3.17 op bladsy 60), en vaar selfs statisties betekenisvol swakker vir TEST. Anders gestel vir Klassifiseerder A en Klassifiseerder B in Tabel 3.16 en Tabel 3.17 op bladsy 60 (dit is TnT teenoor Kombinasie 3) kan die hipotese dat die twee klassifiseerders se foutkoerse nie statisties betekenisvol verskil nie, nie vir UVVM verwerp word nie, met ’n waarskynlikheid dat die verskil tussen die foutkoerse toevallig kan opduik van tot 79.98% vir Presisie-Herroeping – Opsie 6, terwyl daar met redelike sekerheid gesê kan word dat TnT beter vaar as Kombi 3 vir TEST. Kombi 3 toon dus ’n verbetering op enige van die eerstevlakklassifiseerders waaruit dit opgebou is maar vaar so goed soos of slegter as die beste eerstevlakklassifiseerder (TnT) afhangend van die toetsdatastel.

Waar TnT ook deel uitmaak van die kombinasie-woordsoortetiketteerder is die gevolgtrekking dat TnT nie sy hoër akkuraatheid kan handhaaf wanneer dit met drie ander swakker klassifiseerders moet meeding tydens geweegde stemming nie (verwys na Tabel 3.9 op bladsy 55).

In Hoofstuk 1 is die volgende voorwaarde vir ’n kombinasieklassifiseerder gestel, naamlik: ’n Kombinasieklassifiseerder sal beter vaar as enige van die klassifiseerders waaruit dit saamgestel is indien die korrelasie tussen die foute van die onderliggende klassifiseerders klein genoeg is. Anders gestel: die foute van die onderliggende eerstevlakklassifiseerders moet genoeg verskil. Met hierdie stelling in gedagte word ’n verdere gevolgtrekking gemaak uit Tabel 3.3 op bladsy 50 wat die patrone van ooreenkomste tussen die vier eerstevlakklassifiseerders voorhou. As in ag geneem word dat die stemmetode net uit een van die voorgestelde kandidate ’n woordsoortetiket kan toeken, word ’n absolute minimum en maksimum wat deur die kombinasie-woordsoortetiketteerder behaal kan word vasgestel deur die persentasie wat alle eerstevlakklassifiseerders korrek het in te skakel en die persentasie wat alle eerstevlak-klassifiseerders verkeerd het uit te skakel. Hierdie minimum en maksimum word op 62.89% en 88.74% vir UVVM en 72.34% en 94.27% vir TEST onderskeidelik vasgestel. Verder word ’n realistiese maksimum bepaal deur die persentasie waar die korrekte kandidaat deur die minderheid aangewys is ook uit te skakel. Hierdie realistiese maksimum word vasgestel op 81.98% en 90.62% vir UVVM en TEST onderskeidelik. Aanskou die werklike akkuraatheid van die kombinasie-woordsoortetiketteerder (Kombi) tesame met dié van TnT nou in verhouding tot realistiese maksimum (real maks), werklike minimum (min) en werklike maksimum (maks) in Figuur 5.1. Op die grafiese voorstelling kan gesien word dat daar nie veel geleentheid vir verbetering vir Kombi gebied word nie. Die verskil tussen Kombi en real maks is maar 0.90% en 1.30% vir UVVM en TEST onderskeidelik. Hieruit word die gevolgtrekking gemaak dat die korrelasie tussen die foute van die onderliggende klassifiseerders nie klein genoeg is nie en dus nie ’n verbeterde resultaat vir kombinasie tot gevolg het nie.



Figuur 5.1 – Die werklike akkuraatheid van die kombinasie-woordsoortetiketteerder in verhouding tot minimum en maksimum

2. Kan strukturele gedeeltelik-gekontroleerde leer effektief aangewend word om die akkuraatheid van ’n gekontroleerde leer Afrikaanse woordsoortetiketteerder te verbeter?

Die resultate uit afdelings 4.3.5 en 4.3.6 toon dat strukturele gedeeltelik-gekontroleerde leer wel vir sekere gevalle ’n verbeterde resultaat in vergelyking met die gekontroleerde leerbasislyn toon. Op die vraag of dit “effektief aangewend word” is die gevolgtrekking in ag genome die beperkinge van hierdie studie: nee. Alvorens daar oor die redes van hierdie gevolgtrekking uitgebrei word, word daar eers op ’n paar ander sleutelterme uit vraag 2 gefokus.

Strukturele leer is aangewend in die generering van die basislyn gekontroleerde leer woordsoortetiketteerder, sowel as die gedeeltelik-gekontroleerde leer woordsoortetiketteerders. Wanneer die akkuraatheid op TEST van die SVD-ASO gekontroleerde leer woordsoort-etiketteerder (voortaan SVD-ASO basis) in Tabel 4.4 op bladsy 80 met die akkuraathede van die eerstevlakklassifiseerders in Tabel 3.2 op bladsy 50 vergelyk word, is dit noemenswaardig dat SVD-ASO basis, sonder menige toetse en verstellings en die byrekening van menige komplekse eienskapvektore, nie veel slegter vaar nie as die woordsoortetiketteerders, wat met behulp van geoptimaliseerde pakkette wat verskeie eienskappe uit die data in ag neem, afgerig is. Vir SVD-ASO basis **x3** word slegs die kenmerk $Tekseenheid$ in ag geneem tydens afrigting en klassifikasie en dit bereik ’n akkuraatheid van 78.05%. Wanneer die eienskappe $Vorigetekseenheid$ en $Volgendetekseenheid$ hierby gevoeg word (**x1**), word ’n akkuraatheid van 82.77% behaal wat goed vergelyk met die SVM^{light} eerstevlakklassifiseerder in Hoofstuk 3 wat ’n akkuraatheid van 82.38% het.

Gedeeltelik-gekontroleerde leer impliseer die gebruik van ongeëtiketteerde data. Hierdie studie toon dat die gebruik van ongeëttiketteerde data wel ’n verskil kan maak. Vir die geval waar slegs $Vorigetekseenheid$ en $Volgendetekseenheid$ as kenmerk in ag geneem is (**x2**), het die gebruik van ongeëttiketteerde data ’n foutkoersreduksie van 1.67% in Tabel 4.4 op bladsy 80 tot gevolg. Die ander gevalle toon wel ook almal ’n foutkoersreduksie, maar die verskil blyk uit Tabel 4.5 op bladsy 81 nie statisties betekenisvol te wees nie.

Die woord “effektief” impliseer om werksaam en redelik maklik uitvoerbaar te wees. Alhoewel die afrigting ’n veel langer tydsduur as die eerstevlakklassifiseerders uit Hoofstuk 3 vereis (ongeveer 3 dae teenoor ’n paar minute), bly afrigting ’n eenmalige vereiste. Die uiteindelike tydsduur van die klassifikasie van die toetsdatastel neem net ’n minuut of wat. Die prosedure van afrigting is tydsaam, maar verder redelik eenvoudig terwyl dit nie taalspesifieke inmenging of kundigheid verg nie. Verder impliseer “effektief” om doeltreffend te wees, dit is om op die hoogste gekontroleerde leerbasislyn akkuraatheid van 82.77% (Tabel 4.4 op bladsy 80) te

verbeter deur middel van gedeeltelik-gekontroleerde leer. Vir hierdie geval word die doel nie bereik nie, aangesien die positiewe verskil wat daar blyk te wees nie statisties betekenisvol is nie.

Uit die bespreking van die sleutelterme word tot die gevolgtrekking gekom dat strukturele gekontroleerde leer vir sommige gevalle aangewend kan word om die akkuraatheid van ’n gekontroleerde leer Afrikaanse woordsoortetiketteerder te verbeter, maar dat hierdie studie nie daarin slaag om dit *effektief* aan te wend vir alle gevalle nie. Twee van die hoofredes hiervoor is moontlik dat slegs ’n klein persentasie van die beskikbare hulpprobleme uit die data onttrek is vanweë die praktiese uitvoerbaarheid soos bespreek in afdeling 4.3.1, en dat ’n groot verskeidenheid moontlike addisionele eienskappe nie ingesluit is nie.

5.3 VOORUITSKOUING

Die resultate van hierdie studie ontketen onderwerpe vir verdere navorsing:

- Die implementering van ingewikkelder kombinasiemetodes, soos bespreek in afdeling 2.2;
- Die effektiewe aanwending van alle beskikbare hulpbronne soos beskikbaar gestel deur RMA (Eiselen & Puttkammer, 2014) moet ondersoek word deur onder meer gebruik te maak van parallelle programmering;
- Die onttrekking van meer eienskappe in eienskapvektore, byvoorbeeld karaktertipes; karakters aan die begin en einde van tekseenhede; 5-teks-eenheid venster rondom die huidige tekseenheidposisie; klassifikasie van twee tekseenhede links van die huidige posisie; bi-gramme van die huidige tekseenheid en die klassifikasie links van die huidige tekseenheid; klassifikasies wat voorheen aan die huidige tekseenheid toegeken is (Ando & Zhang, 2005a);
- Die toepassing van die tegnieke op ander taaltegnologiese toepassings; en
- Die toepassing van die tegnieke op ander hulpbronskaarstale.

5.4 SLOT

Alhoewel die eksperimente met betrekking tot kombinasie en strukturele gedeeltelik-gekontroleerde leer toon dat dit wel moontlik is om die akkuraatheid vir sommige gevalle te verhoog, slaag hierdie studie, wat weens praktiese uitvoerbaarheid aan aansienlike beperkings in terme van omvang van gebruikte hulpbronne en eienskapvektore onderhewig is, nie daarin

om ’n effektiewe verbetering vir die toonaangewende gevalle teweeg te bring nie. Dit bevestig die uitdaging wat steeds bestaan vir hulpbronskaars tale in die veld van Mensetaaltegnologie.

BRONNELYS

Abney, S. 2007. Semi-supervised learning for computational linguistics. University of Michigan: CRC Press.

Ando, R.K. & Zhang, T. 2005a. A framework for learning predictive structures from multiple tasks and unlabelled data. *The Journal of Machine Learning Research*, 6:1817-1853.

Ando, R.K. & Zhang, T. 2005b. A high-performance semi-supervised learning method for text chunking. (In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Bacchiani, M., Riley, M., Roark, B. & Sproat, R. 2006. MAP adaptation of stochastic grammars. *Computer speech & language*, 20(1):41-68.

Belkin, M., Niyogi, P. & Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399-2434.

Blitzer, J. & Zhu, X.J. 2008. Semi-supervised learning for natural language processing. (In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Blum, A. & Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincuts. Carnegie Mellon University.

Blum, A. & Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. (In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* organised by ACM).

Bouma, G., Van Noord, G. & Malouf, R. 2001. Alpino: Wide-coverage computational analysis of Dutch. *Language and Computers*, 37(1):45-59.

Brants, T. 2000. TnT: a statistical part-of-speech tagger. (In *Proceedings of the Sixth Conference on Applied Natural Language Processing* organised by Association for Computational Linguistics).

Breiman, L. 1996. Bagging predictors. *Machine learning*, 24(2):123-140.

Brill, E. 1994. Some advances in transformation-based part of speech tagging. (In *Proceedings of the Twelfth National Conference on Artificial Intelligence*-Volume 1).

Brill, E. & Wu, J. 1998. Classifier combination for improved lexical disambiguation. (In *Proceedings of the 17th International Conference on Computational Linguistics*-Volume 1 organised by Association for Computational Linguistics).

Buchholz, S. & Marsi, E. 2006. CoNLL-X shared task on multilingual dependency parsing. (In *Proceedings of the Tenth Conference on Computational Natural Language Learning* organised by Association for Computational Linguistics).

Burnard, L. Reference guide for the British National Corpus (XML edition), 2007. http://www.natcorp.ox.ac.uk/docs/URG Date of access: 15 Apr. 2015.

Caruana, R. 1997. Multitask Learning. *The Journal of Machine Learning – Special issue on inductive transfer*, 28(1):41-75.

Chapelle, O., Schölkopf, B. & Zien, A. 2006. Semi-supervised learning. Cambridge, Massachusetts: The MIT Press.

Charniak, E. 1997. Statistical parsing with a context-free grammar and word statistics. *AAAI/IAAI*, 2005:598-603.

Clark, S., Curran, J.R. & Osborne, M. 2003. Bootstrapping POS taggers using unlabelled data. (In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*-Volume 4 organised by Association for Computational Linguistics).

Cole, R.A., Varile, G.B. & Zampolli, A. 1997. Survey of the state of the art in human language technology. Cambridge University Press.

The British National Corpus, version 3 (BNC XML Edition). 2007. Distributed by Oxford University Computing Services on behalf of the BNC Consortium. http://www.natcorp.ox.ac.uk/ Date of access: 15 Apr. 2015.

Cozman, F.G., Cohen, I. & Cirelo, M.C. 2003. Semi-supervised learning of mixture models. (In *ICML 2003,* p. 99-106).

Daelemans, W. & Osborne, M. 2003. (In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*-Volume 4 organised by Association for Computational Linguistics).

Daelemans, W., Van Den Bosch, A. & Weijters, T. 1997. IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):407-423.

Daelemans, W. & Zajac, R. 2001. (In *Proceedings of the 2001 Workshop on Computational Natural Language Learning*-Volume 7 organised Association for Computational Linguistics).

Daelemans, W., Zavrel, J., Berck, P. & Gillis, S. 1996. MBT: A memory-based part of speech tagger-generator. arXiv preprint cmp-lg/9607012.

Daelemans, W., Zavrel, J., Van der Sloot, K. & Van den Bosch, A. 2003. Timbl: Tilburg memory-based learner. *ILK* 03, 10.

Dasgupta, S., Littman, M.L. & McAllester, D. 2002. PAC generalization bounds for co-training. Advances in neural information processing systems, 1:375-382.

Dasgupta, S. & Ng, V. 2007. Unsupervised Part-of-Speech Acquisition for Resource-Scarce Languages. (In *EMNLP-CoNLL* 2007).

Davis, J. & Goadrich, M. 2006. The relationship between Precision-Recall and ROC curves. (In *Proceedings of the 23rd International Conference on Machine Learning* organised by ACM).

Dayan, P. 1999. Unsupervised learning. The MIT encyclopaedia of the cognitive sciences.

Dietterich, T.G. 1997. Machine learning research. *Artificial Intelligence Magazine*, 18(4):97.

Dietterich, T.G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895-1923.

Efron, B. & Tibshirani, R.J. 1994. An introduction to the bootstrap. Chapman and Hall: CRC press.

Eiselen, E.R. & Puttkammer, M.J. 2014. Developing Text Resources for Ten South African Languages. (In *Proceedings of the 9th International Conference on Language Resources and Evaluation*. Reykjavik, Iceland. p. 3698-3703).

Elkan, C. 2011. Nearest neighbor classification. *Technical report 2011.* elkan@cs.ucsd.edu.

Escudero, G., Màrquez, L. & Rigau, G. 2000. Boosting applied to word sense disambiguation. Springer.

Evgeniou, A. & Pontil, M. 2007. Multi-task feature learning. *Advances in neural information processing systems*, 19:41.

Fodor, I.K. 2002. A survey of dimension reduction techniques: *Technical Report* UCRL-ID-148494, Lawrence Livermore National Laboratory.

Francis, W.N. & Kucera, H. 1979. Brown Corpus manual: Manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. Providence, Rhode Island, USA: Brown University.

Frederking, R. & Nirenburg, S. 1994. Three heads are better than one. (In *Proceedings of the Fourth Conference on Applied Natural Language Processing* organised by Association for Computational Linguistics).

Freund, Y. & Schapire, R.E. 1996. Experiments with a new boosting algorithm. (In *ICML 1996*, 96:148-156).

Fujino, A., Ueda, N. & Saito, K. 2005. A hybrid generative/discriminative approach to semi-supervised classifier design. (In *Proceedings of the National Conference on Artificial Intelligence*).

Garside, R. & Smith, N. 1997. A hybrid grammatical tagger: *CLAWS4. Corpus annotation: Linguistic information from computer text corpora*:102-121.

Ghahramani, Z. 2004. Unsupervised learning. (In *Advanced Lectures on Machine Learning*. Springer. p. 72-112).

Grira, N., Crucianu, M. & Boujemaa, N. 2004. Unsupervised and semi-supervised clustering: a brief survey. A review of machine learning techniques for processing multimedia content, Report of the MUSCLE European Network of Excellence (FP6).

Haruechaiyasak, C. 2008. A Tutorial on Naive Bayes Classification. *Technical Report*. http://suanpalm3.kmutnb.ac.th/teacher/FileDL/choochart82255418560.pdf.

Henderson, J.C. & Brill, E. 1998. Exploiting diversity for natural language processing. (In *AAAI/IAAI.* p.1174).

Henderson, J.C. & Brill, E. 2000. Bagging and boosting a treebank parser. (In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference* organised by Association for Computational Linguistics).

Hoste, V. & Daelemans, W. 2000. Comparing bagging and boosting for natural language processing tasks: a typicality approach. (In *Proceedings of the Tenth Belgian Dutch Conference on Machine Learning*, BENELEARN 2000 organised by Citeseer).

Huang, G., Song, S., Gupta, J.N. & Wu, C. 2014. Semi-supervised and unsupervised extreme learning machines. (In *Cybernetics, IEEE Transactions*, 44(12):2405-2417).

Joachims, T. 1999a. SVM-Light Support Vector Machine http://svmlight. joachims.org/, University of Dortmund, 19(4). Date of access: 15 Apr. 2015.

Joachims, T. 1999b. Transductive inference for text classification using support vector machines. (In *ICML* 99:200-209).

Jones, R. 2005. Learning to extract entities from labeled and unlabeled text. Utah: University of Utah. (Thesis – PhD).

Jurafsky, D. & Martin, J.H. 2000. Speech & language processing. India: Pearson Education.

Karahan, M., Hakkani-Tur, D., Riccardi, G. & Tur, G. 2003. Combining classifiers for spoken language understanding. (In *Automatic Speech Recognition and Understanding*, 2003. ASRU'03. 2003 IEEE Workshop, IEEE).

Klein, D. & Manning, C.D. 2002. A generative constituent-context model for improved grammar induction. (In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Kudo, T. & Matsumoto, Y. 2001. Chunking with support vector machines. (In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies* organised by Association for Computational Linguistics).

Lewis, M.P., Simons, G.F. & Fennig, C.D. 2015. Ethnologue: Languages of the World. 18th edition. Dallas, USA: SIL International.

Lin, X., Yacoub, S., Burns, J. & Simske, S. 2003. Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24(12):1959-1969.

Marquez, L. 2000. Machine learning and natural language processing (In *Proceedings for Conference "Aprenidizaje automático aplicado al procesamiento del lenguaje natural*").

McCarthy, J. & Feigenbaum, E.A. 1990. In Memoriam: Arthur Samuel: Pioneer in Machine Learning. *AI Magazine*, 11(3):10.

Mitchell, T.M. 1997. Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 45.

Ng, A. 2000. CS229 Lecture notes. *CS229 Lecture notes*, 1(1):1-3.

Nigam, K., McCallum, A.K., Thrun, S. & Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2-3):103-134.

Niu, Z.-Y., Ji, D.-H. & Tan, C.L. 2005. Word sense disambiguation using label propagation based semi-supervised learning. (In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Pang, B. & Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. (In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Peché, M. 2009. Spoken language identification in resource-scarce environments. Pretoria: University of Pretoria. (Thesis – PhD).

Pilon, S. 2005. Outomatiese Afrikaanse Woordsoortetikettering. Verhandeling voorgelê vir die graad Magister Artium in Algemene Taal en Literatuurwetenskap aan die Noordwes-Universiteit, Potchefstroomkampus.

Poel, M., Stegeman, L. & op Den Akker, R. 2007. A support vector machine approach to Dutch part-of-speech tagging. (In *Advances in Intelligent Data Analysis* VII. Springer. p. 274-283).

Rajaraman, A. & Ullman, J.D. 2011. Mining of massive datasets. Cambridge University Press.

Ramshaw, L.A. & Marcus, M.P. 1995. Text chunking using transformation-based learning. (In *Natural language processing using very large corpora.* Springer. p. 157-176).

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. (In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* organised by Philadelphia, PA).

Sampson, G. 2003. The LUCY Corpus: Documentation. Sussex: University of Sussex.

Samuel, A.L. 2000. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 44(1.2):206-226.

Sang, E.F., Daelemans, W., Déjean, H., Koeling, R., Krymolowski, Y., Punyakanok, V., et al. 2000. Applying system combination to base noun phrase identification. (In *Proceedings of the 18th Conference on Computational linguistics*-Volume 2 organised by Association for Computational Linguistics).

Sang, E.F. & Déjean, H. 2001. Introduction to the CoNLL-2001 shared task: clause identification. (In *Proceedings of the 2001 Workshop on Computational Natural Language Learning*-Volume 7 organised by Association for Computational Linguistics).

Sang, E.F. & Veenstra, J. 1999. Representing text chunks. (In Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics organised by Association for Computational Linguistics).

Sarkar, A. 2001. Applying co-training methods to statistical parsing. (In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies* organised by Association for Computational Linguistics).

Schapire, R. 2003a. Introduction, basic definitions, the consistency model. *COS 511: Foundations of Machine Learning*. http://www.cs.princeton.edu/courses/archive/spr03/cs511/index.html. Date of access: 15 Apr. 2015.

Schapire, R.E. 2003b. The boosting approach to machine learning: An overview. (*In* Hansen, M., Holmes, C.C., Mallick, B.K., & Yu, B. *eds*. Nonlinear estimation and classification. Springer. p. 149-171).

Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., et al. 2003. Bootstrapping statistical parsers from small datasets. (In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*-Volume 1 organised by Association for Computational Linguistics).

Subramanya, A., Petrov, S. & Pereira, F. 2010. Efficient graph-based semi-supervised learning of structured tagging models. (In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* organised by Association for Computational Linguistics).

Subramanya, A. & Talukdar, P.P. 2012. Graph-based semi-supervised learning algorithms for NLP. (In *Tutorial Abstracts of ACL 2012* organised by Association for Computational Linguistics).

Nederlandse Taalunie. 2011. STEVIN: Spraak-en Taaltechnologische Essentiële Voorzieningen in het Nederlands. http://taalunie.org/organisatie/netwerk/stevin Date of access: 20 Apr. 2015.

Tjong Kim Sang, E.F. & Buchholz, S. 2000. Introduction to the CoNLL-2000 shared task: Chunking. (In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*-Volume 7 organised by Association for Computational Linguistics).

Trushkina, J. 2004. Morpho-syntactic annotation and dependency parsing of German. Tübingen: University Tübingen. (Thesis – PhD).

Utsuro, T., Sassano, M. & Uchimoto, K. 2002. Combining outputs of multiple Japanese named entity chunkers by stacking. (In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*-Volume 10 organised by Association for Computational Linguistics).

Van Halteren, H. 1999. Weighted probability distribution voting, an introduction. (In *CLIN 1999*).

Van Halteren, H., Zavrel, J. & Daelemans, W. 1998. Improving data driven wordclass tagging by system combination. (In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Van Halteren, H., Zavrel, J. & Daelemans, W. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199-229.

Van Huyssteen, G.B. 2003. Preface – Human Language Technology in South Africa: Resources and Applications. *Southern African Linguistics and Applied Language Studies*, 21(3): iii-v.

Vapnik, V. 2000. The nature of statistical learning theory. Springer Science & Business Media.

Vapnik, V.N. 1998. Statistical learning theory. Vol. 1: Wiley New York.

Veenstra, J. 1999. Memory-based text chunking. (In *Machine learning in human language technology*. Workshop at ACAI organised by Citeseer).

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivalling supervised methods. (In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics on Human Language Technologies* organised by Association for Computational Linguistics).

Zhang, T. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. (In *Proceedings of the Twenty-first International Conference on Machine Learning* organised by ACM).

Zhu, X. 2005. Semi-supervised learning literature survey. (Unpublished).

Zhu, X. 2009. Semi-supervised learning literature survey. (Unpublished).

Zhu, X. & Goldberg, A.B. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1-130.

BYLAAG A: ENGELSE TERME

’n Lys van die relevante Engelse terme wat in hierdie studie gebruik word.

Engelse term	**Afrikaanse term**
alternating structure optimisation	wisselstruktuuroptimering (ASO)
auxiliary problems	hulpprobleme
bagging	groepering
beginning	begin
bias correction	sydigheidskorreksie
boosting	versterking
bootstrap sub samples	skoenlussteekproewe
bootstrapping	skoenlusproses / skoenlussteekproefneming
British National Corpus	BNC
call classification	gesproke taal
cascading	kaskadekombinasiemetode
case base	klassifikasiegevalbasis
chunker	sinstukverdeler
chunking	sinstukverdeling
clause identification	sinsnede-identifisering
clustering	trosvorming
co-training	deelleer
decision boundary	beslissingsgrens
deleted interpolation	geskrapte interpolasie
dimensionality reduction	dimensionaliteitvermindering
discriminative training	diskriminerende afrigting
Empirical Risk Minimization	Empiriese Risiko Minimalisering
enabling technology	kerntegnologie
end	einde
error rate	foutkoers
error rate reduction	foutkoersreduksie
factor analysis	faktoranalise
false negatives	vals negatiewe
false positives	vals positiewe

Gaussian Mixture Models	Gauss-mengselmodelle (GMM)
generative models	generatiewe modelle
graph-based methods	grafiekgebaseerdemetodes
harmonic function	harmoniese funksie
hat loss function	hoedverliesfunksie
Hidden Markov Models (HMM)	versteekte Markov-modelle
hinge loss function	skarnierverliesfunksie
Human Language Technology	Mensetaaltegnologie
independent components analysis	onafhanklikekomponente-analise
information gain	inligtingswins
inside	binne
k-means	k-gemiddeldes
lazy boosting	trae versterking
learning tasks	leertake
lemmatisation	lemma-identifisering
linearly separable	lineêr skeibaar
low-typical cases	nie-tipiese gevalle
majority voting	meerderheidstemming
manifold regularisation	veelvuldige regulering
maximum likelihood estimate (MLE)	maksimumaanneemlikheidsberaming
memory-based tagger generation and tagging	geheuegebaseerde etiketteringsgenerasie en etikettering
mixture of Gaussians	mengsel van Gausse
morphological decomposition	morfologiese ontbinding
morpho-syntactic tagging	morfo-sintaktiese etikettering
Multinomial Mixture Model	multinomialemengselmodel
multiple prediction problems	meervoudige klassifikasietake
multilingual dependency parser	meertalige sinsontleder
multi-task learning	multi-taak leer
named entity extraction	tekseenheidonttrekking
noun phrase chunking	klassifikasie van naamwoordstukke
outside	buite
parse	sinsontleding
parser switching	sinsontledingomruiling

parsing	sinsontleding
precision	presisie
prediction error	voorspellingsfout
prediction space, a set of predictors	voorspellingsruimte
predictions	voorspellings
plurality voting	pluraliteitstemming
principal components analysis (PCA)	hoofkomponente-analise
putative label	veronderstelde annotasie
recall	herroeping
regulariser	reguleerder
Resource Management Agency	hulpbronbestuursagentskap
resource-scarce languages	hulpbronskaars tale
sample points of the predictor distribution in the predictor space	steekproefpunte van die voorspellerverdeling in die voorspellingsruimte
self-training	selfleer
semi-supervised Support Vector Machines	gedeeltelik-gekontroleerde steunvektorklassifiseerders (S3VM)
signed distance	betekende afstand
singular value decomposition	singulierewaarde-ontbinding (SVD)
slack variables	spelingsveranderlikes
smooth	glad
smoothing	gladstryking
stacked probabilistic voting	opgestapelde waarskynlikheidstemming
stacking	opstapeling
stochastic gradient descent	stogastiese gradiëntafdaling (SGD)
structural learning	strukturele leer
successive abstraction	opeenvolgende abstraksie
Support Vector Machines	steunvektorklassifiseerders (SVM)
System Internal Combination	interne kombinasie
target problem	teikenprobleem
Tilburg Memory-Based Learner	TiMBL
total precision	totale akkuraatheid
Trigrams'n'Tags	TnT
true negatives	werklike negatiewe

true positives	werklike positiewe
tune data	insteldata
typical cases	tipiese gevalle
weighted probability distribution voting	geweegde waarskynlikheidsverdelingstemming (WPDV)
weighted voting	geweegde stemming
word sense disambiguation	woordbetekenisvereenduidiging

BYLAAG B: WOORDSOORTETIKETTE

Afrikaanse woordsoortetikette met beskrywings

Etiket	Beskrywing
AOA	Adjektief: oortreffend/attributief
AOP	Adjektief: oortreffend/predikatief
ASA	Adjektief: stellend/attributief
ASP	Adjektief: stellend/predikatief
AVA	Adjektief: vergrotend/attributief
AVP	Adjektief: vergrotend/predikatief
BOM	Bywoord: oortreffend/maat
BOW	Bywoord: oortreffend/ wyse
BSD	Bywoord: stellend/modaliteit
BSF	Bywoord: stellend/frekwensie
BSG	Bywoord: stellend/graad
BSM	Bywoord: stellend/maat
BSO	Bywoord: stellend/ontkenning
BSP	Bywoord: stellend/plek
BSR	Bywoord: stellend/rigting
BSS	Bywoord: stellend/skakeling
BST	Bywoord: stellend/tyd
BSW	Bywoord: stellend/wyse
BVM	Bywoord: vergrotend/maat
BVW	Bywoord: vergrotend/ wyse
KN	Konjunk: neweskikkend
KO	Konjunk: onderskikkend
LB	Lidwoord: bepaald
LO	Lidwoord: onbepaald
NA	Naamwoord: Abstrak
NEE	Naamwoord: eienaam/enkelvoud/basis
NEM	Naamwoord: eienaam/meervoud/basis
NLW	Naamwoord: eienaam/meervoud/diminutief
NM	Naamwoord: Massanaam
NME	Naamwoord: maatnaam/enkelvoud/basis
NMED	Naamwoord: maatnaam/enkelvoud/diminutief
NSE	Naamwoord: soortnaam/enkelvoud/basis
NSED	Naamwoord: soortnaam/enkelvoud/diminutief
NSM	Naamwoord: soortnaam/meervoud/basis
NSMD	Naamwoord: soortnaam/meervoud/diminutief
NVE	Naamwoord: versamelnaam/enkelvoud/basis
NVED	Naamwoord: versamelnaam/enkelvoud/diminutief
NVM	Naamwoord: versamelnaam/meervoud/basis
PA	Voornaamwoord: Aanwysend
PB	Voornaamwoord: betreklik
PDHEB	Voornaamwoord: derde/enkelvoud/wederkerend

PDHEDP	Voornaamwoord: derde/enkelvoud/wederkerend
PDHENP	Voornaamwoord: derde/enkelvoud/wederkerend
PDHEW	Voornaamwoord: derde/enkelvoud/wederkerend
PDMB	Voornaamwoord: derde/meervoud/besitlik
PDMP	Voornaamwoord: derde/meervoud/persoonlik
PDMW	derde/meervoud/wederkerend
PDOENP	derde/meervoud/wederkerend
PDOEP	derde/meervoud/wederkerend
PDOEW	derde/meervoud/wederkerend
PDVEB	derde/meervoud/wederkerend
PDVEDP	derde/meervoud/wederkerend
PDVENP	derde/meervoud/wederkerend
PDVEW	derde/meervoud/wederkerend
PEEB	Voornaamwoord: eerste/enkelvoud/besitlik
PEEDP	Voornaamwoord: eerste/enkelvoud/gemarkeerd/persoonlik
PEENP	Voornaamwoord: eerste/enkelvoud/ongemarkeerd /persoonlik
PEEW	Voornaamwoord: eerste/enkelvoud/wederkerend
PEMB	Voornaamwoord: eerste/meervoud/besitlik
PEMNP	Voornaamwoord: eerste/meervoud/persoonlik
PEMP	Voornaamwoord: eerste/meervoud/persoonlik
PEMW	Voornaamwoord: eerste/meervoud/wederkerend
PO	Voornaamwoord: onbepaald
PR	Voornaamwoord: onbepaald
PTEB	Voornaamwoord: tweede/enkelvoud/besitlik
PTEDP	Voornaamwoord: tweede/enkelvoud/gemarkeerd /persoonlik
PTENP	Voornaamwoord: tweede/enkelvoud/ongemarkeerd /persoonlik
PTEW	Voornaamwoord: tweede/enkelvoud/wederkerend
PTMB	Voornaamwoord: tweede/meervoud/besitlik
PTMNP	Voornaamwoord: tweede/meervoud/persoonlik
PTMP	Voornaamwoord: tweede/meervoud/persoonlik
PV	Voornaamwoord: Vraend
PW	Voornaamwoord: wederkerig
PX	Voornaamwoord: wederkerig
RAE	Residu: afkorting/enkelvoud
RKE	Residu: akroniem/enkelvoud
RLE	Residu: letternaamwoord/enkelvoud
RLM	Residu: letternaamwoord/meervoud
RSE	Residu: simbool/enkelvoud
RSF	Residu: suffiks
RVE	Residu: vreemdetaalwoord/enkelvoud
RVM	Residu: vreemdetaalwoord/meervoud
RWD	Residu: woorddeel
SVS	Setsel: voorsetsel
THAB	Telwoord: hooftelwoord/adjektief/bepaald
THAO	Telwoord: hooftelwoord/adjektief/onbepaald
THBB	Telwoord: hooftelwoord/bywoord/bepaald
THBO	Telwoord: hooftelwoord/bywoord/onbepaald

THNB	Telwoord: hooftelwoord/naamwoord/bepaald
THPB	Telwoord: hooftelwoord/voornaamwoord/bepaald
THPO	Telwoord: hooftelwoord/voornaamwoord/onbepaald
TRAB	Telwoord: rangtelwoord/adjektief/bepaald
TRAO	Telwoord: rangtelwoord/adjektief/onbepaald
TRBB	Telwoord: rangtelwoord/bywoord/bepaald
TRNB	Telwoord: rangtelwoord/bywoord/onbepaald
TRPB	Telwoord: rangtelwoord/voornaamwoord/bepaald
UDS	Uniek: dis
UE	Uniek: enklities
UPB	partikel/betreklik
UPD	Uniek: partikel/deel
UPE	Uniek: partikel/skakel
UPG	Uniek: partikel/graad
UPI	Uniek: partikel/infinitief
UPO	Uniek: partikel/ontkenning
UPS	Uniek: partikel/genitief
UPV	Uniek: partikel/vergelyking
UPW	Uniek: partikel/ww.
UXD	Uniek: eksistensiële daar
VNAHG	Uniek: eksistensiële daar
VTHOG	Werkwoord: teenwoordig/hoof/onskeibaar/oorganklik
VTHOK	Werkwoord: teenwoordig/hoof/onskeibaar/koppel
VTHOO	Werkwoord: teenwoordig/hoof/onskeibaar/onoorganklik
VTHOV	Werkwoord: teenwoordig/hoof/onskeibaar/voorsetsel
VTHSG	Werkwoord: teenwoordig/hoof/skeibaar/oorganklik
VTHSK	Werkwoord: teenwoordig/hoof/skeibaar/oorganklik
VTHSO	Werkwoord: teenwoordig/hoof/skeibaar/onoorganklik
VTUOA	Werkwoord: teenwoordig/hulp/onskeibaar/aspek
VTUOM	Werkwoord: teenwoordig/hulp/onskeibaar/modaliteit
VTUOP	Werkwoord: teenwoordig/hulp/onskeibaar/modus
VUOT	Werkwoord: hulp/onskeibaar/tyd
VVHOG	Werkwoord: verlede/hoof/onskeibaar/oorganklik
VVHOK	Werkwoord: verlede/hoof/onskeibaar/koppel
VVHOO	Werkwoord: verlede/hoof/onskeibaar/onoorganklik
VVHOV	Werkwoord: verlede/hoof/onskeibaar/voorsetsel
VVUOA	Werkwoord: verlede/hulp/onskeibaar/aspek
VVUOM	Werkwoord: verlede/hulp/onskeibaar/modaliteit
VVUOP	Werkwoord: verlede/hulp/onskeibaar/modus
W	Tussenwerpsel
ZE	Punktuasie: sinseinde
ZM	Punktuasie: sinseinde
ZPL	Punktuasie: links-parentese
ZPR	Punktuasie: regs-parentese

BYLAAG C: 1000 MEES FREKWENTE TEKSEENHEDE

Die 1000 mees frekwente tekseenhede soos onttrek uit die ongeëtiketteerde data

Tekseenheid	**Frekwensie**
die	8451861
.	7458310
,	5871189
van	3802039
het	3286366
en	3033161
'n	2963398
in	2942692
is	2645361
nie	1736641
"	1450339
te	1397652
wat	1389158
Die	1313738
)	1267968
(	1259218
om	1204581
op	1178935
se	1112871
met	1063870
word	1045670
vir	1031483
sy	949104
dat	853397
:	722237
sal	704859
aan	695813
as	640147
hy	638548
was	592295
dit	578047
kan	544661
by	487312
oor	468156
ook	468118
gesê	464567
deur	457300
tot	420839
hulle	386860

na	379776
;	375303
dié	374479
of	357964
Dit	338618
mnr.	335392
moet	328950
jaar	323900
maar	322151
hul	298358
Hy	288735
teen	281304
gaan	277194
meer	258229
uit	258063
sê	256364
daar	252799
ons	250001
wees	233949
twee	233199
gister	225221
nog	224247
net	221646
haar	218932
toe	216455
hom	213689
Suid-Afrika	209400
een	204629
mense	199188
Mnr.	199058
ander	192863
baie	191472
soos	190290
eerste	188050
nuwe	180390
ek	174909
In	170133
onder	169746
nou	168756
al	168214
'	158657
groot	157651
verlede	153173
begin	148262
Ek	144778
maak	141177
ná	138836

wil	136496
egter	136483
my	136334
weer	135243
so	130239
tussen	127743
reeds	127653
drie	126067
voor	124618
Volgens	123717
nadat	121544
Sy	121347
laat	120838
miljoen	120415
waar	120196
omdat	118403
sowat	117388
kry	114835
Dié	113973
1	113276
Ons	109212
doen	109020
000	107603
kom	106224
land	103885
2	103599
Daar	103007
Van	102047
vanjaar	100275
kon	99783
Pretoria	99585
werk	99398
gebruik	98389
jy	96548
Suid-Afrikaanse	94651
?	93344
keer	92746
sou	91565
geen	91467
gemaak	90549
saam	89276
Hulle	88000
per	84698
hier	83845
terwyl	83571
span	83189
gedoen	83152

hoe	82472
the	81945
besluit	81517
tyd	78845
speel	78706
hou	78671
3	77197
deel	76941
dag	76531
mens	76432
jou	76148
DIE	75678
Maar	75265
As	74648
ANC	73283
tweede	73207
man	72983
geld	72908
SA	72544
goeie	70502
goed	70351
vier	69324
glo	67990
groep	67631
bekend	67099
af	66702
de	66594
stel	66581
week	66575
Polisie	65727
En	65557
veral	65502
gee	64854
steeds	64118
hierdie	63391
alle	63120
Johannesburg	62974
+	62582
staan	62569
einde	62242
moes	61748
dan	60847
dr.	60781
weens	60580
huis	60568
maande	60510
vrou	60292

eie	59689
gehad	59649
sien	59158
elke	58383
punte	58345
geneem	58333
wedstryd	58204
wanneer	58052
hê	58045
laaste	57972
plek	57550
regering	57302
vandag	57272
ondersoek	56799
der	56583
neem	56500
paar	55697
afgelope	55469
kinders	55343
4	54793
dood	54752
beste	54728
De	54244
weet	53452
daarvan	53280
Amerikaanse	52244
wêreld	51725
bly	51556
vyf	51550
!	51456
waarin	51405
later	51233
volgende	50609
sonder	50579
5	50373
wie	49617
lyk	49543
Datum	49449
lede	49400
Saterdag	49332
eers	49304
moontlik	49171
probeer	49085
verwag	49084
dieselfde	48838
lank	48789
gewen	47983

politieke	47962
ten	47957
lewe	47939
gekry	47931
verskeie	47359
dae	46866
grootste	46707
vorige	46675
mev.	46642
ontvang	46537
ses	46505
swart	46400
volgens	46358
betaal	46259
Afrika	46008
voordat	45701
gehou	45614
maatskappy	45595
Amerika	45573
Met	45421
Beeld	45241
sterk	45217
beter	45105
enige	45103
internasionale	44720
wou	44245
voorsitter	44240
gelede	43989
selfs	43845
plaaslike	43635
spelers	43480
mans	43341
meeste	43249
sake	43146
betrokke	43020
meen	42879
$	42854
6	42804
saak	42428
want	42402
help	42378
almal	42323
Op	42149
sedert	42122
10	41728
behoort	41610
reg	40971

aanstaande	40946
Vir	40883
gespeel	40495
persent	40247
só	40196
pres.	40066
teenoor	40050
wel	40026
wen	40008
bestaan	39904
bied	39806
gestel	39736
verkoop	39709
jare	39663
binne	39521
gegee	39372
nasionale	39343
agter	39306
Regering	39203
vroue	39073
derde	39020
dus	38946
daarop	38602
behaal	38535
iets	38446
maand	38432
mark	38371
Kaapstad	38062
kort	37916
sluit	37873
hele	37723
Mandela	37345
minder	37304
motor	37268
nodig	37230
probleme	37187
dink	37136
plaas	36966
self	36933
vroeër	36823
beskikbaar	36630
nooit	36485
rol	36411
direkteur	36341
raak	36180
departement	35965
hoër	35958

aandele	35903
staat	35862
inligting	35825
geval	35814
lande	35765
pad	35740
ter	35690
Nasionale	35638
VSA	35623
verloor	35316
asook	35284
mag	35249
ekonomiese	35235
gebeur	35139
prys	35040
beteken	35039
naweek	35020
Transvaal	34951
7	34937
dalk	34891
Outeur	34639
min	34590
besoek	34587
waarskynlik	34472
gestyg	34269
gebied	34202
Dr.	34180
raad	34174
reeks	34129
Wat	34092
veel	34001
0	33994
verklaring	33862
gevra	33821
besig	33769
Afrikaanse	33673
bereik	33627
kyk	33448
genoeg	33343
Maart	33273
bring	33175
Hoewel	33112
skryf	33092
Britse	33057
John	32975
ou	32588
aanvaar	32557

geword	32340
jong	32283
8	32228
Dis	32114
Botha	32097
gekom	32079
groter	31987
verskyn	31986
bestuur	31898
sodat	31887
gevolg	31843
altyd	31837
hof	31586
huidige	31577
miljard	31509
alles	31463
pas	31254
druk	31250
mekaar	31112
terug	31112
Johan	30976
buite	30913
klein	30833
du	30832
groei	30781
praat	30775
daardie	30729
daarna	30571
Vrydag	30476
hoof	30434
onlangs	30351
verder	30174
weke	30136
skool	30125
beskou	30092
tien	29974
naam	29913
naby	29579
Een	29579
polisie	29492
gesien	29407
Afrikaans	29359
uitvoerende	29347
vind	29344
kos	29262
aangebied	29176
lewer	29030

Natal	28972
lê	28889
20	28858
waarde	28833
buitelandse	28787
verander	28678
Toe	28677
beslis	28512
byna	28477
verkiesing	28421
niks	28412
Om	28400
sit	28080
Februarie	28016
9	27962
hoë	27957
albei	27949
eergister	27943
finansiële	27923
verbeter	27895
April	27834
Suid	27815
sekere	27733
toekoms	27572
15	27536
wys	27520
vraag	27512
Klerk	27369
spel	27347
kans	27345
jongste	27344
Universiteit	27290
tans	27262
berig	27250
enigste	27196
laer	27194
belang	27051
vergadering	27048
woordvoerder	27017
beheer	26881
beurt	26867
vertel	26771
gesluit	26767
koop	26759
volg	26744
duidelik	26615
12	26613

Desember	26411
hoop	26311
diens	26193
maatskappye	26156
voormalige	25969
Mev.	25950
seker	25946
water	25920
grond	25772
Al	25666
lig	25639
verseker	25572
verslag	25488
vra	25355
aandeel	25340
FOTO	25304
30	25301
slegs	25266
program	25234
ontwikkeling	25211
probleem	25181
weg	25169
waarop	25161
bedryf	25126
Só	25112
uur	25002
pryse	24980
stryd	24927
toets	24914
spanne	24886
Nog	24789
verhoog	24735
ekonomie	24690
September	24677
vlak	24507
25	24489
tydens	24456
&	24456
Engeland	24435
minister	24415
feit	24374
Januarie	24360
lei	24358
Junie	24343
lopies	24324
Durban	24141
gebring	23919

vandeesweek	23848
Oktober	23807
Australië	23792
Sapa	23719
iemand	23684
Ná	23659
Johannesburgse	23640
party	23515
belangrike	23510
geleentheid	23469
duur	23387
David	23381
seun	23348
Minister	23303
indien	23303
wedstryde	23288
Chris	23198
ronde	23183
Geen	23143
geweld	23136
voel	23091
Twee	22963
môre	22948
100	22898
manier	22856
skole	22746
regte	22679
loop	22614
Jan	22575
u	22559
soek	22549
steun	22541
wag	22490
verskillende	22476
gevind	22462
16	22355
Mei	22325
byvoorbeeld	22104
private	22059
president	22048
volle	22000
gisteraand	21952
eerder	21907
Mark	21879
Vrystaat	21823
sewe	21821
swak	21810

sulke	21759
gedaal	21746
NP	21725
011	21717
November	21714
ma	21705
me.	21670
koste	21648
komitee	21626
Suid-Afrikaners	21615
wit	21571
Smith	21559
musiek	21519
Londen	21514
Net	21511
18	21496
11	21495
inwoners	21485
openbare	21473
lang	21453
tree	21453
lid	21445
hegtenis	21436
bevind	21407
Nuwe	21392
vol	21369
gemeenskap	21352
13	21329
bepaal	21310
HET	21209
pa	21153
aandag	21132
verskaf	21086
maklik	21067
waarvan	21058
gebou	21044
14	21034
helfte	21018
anders	20963
deelneem	20962
Julie	20945
dra	20871
dikwels	20853
24	20847
skep	20842
ontwikkel	20812
André	20805

Nou	20798
Merwe	20793
stem	20786
elk	20781
dienste	20753
vorm	20747
sommige	20713
aangekondig	20698
seisoen	20693
bespreek	20563
stad	20543
afrigter	20518
minute	20481
besef	20436
Augustus	20430
studente	20408
getuig	20378
New	20372
Deur	20350
gehoor	20336
rede	20324
WP	20297
Michael	20247
kind	20218
verband	20217
agt	20195
kop	20194
dinge	20084
skuld	20073
Brittanje	20072
oorwinning	19992
besit	19977
Engelse	19895
handel	19874
daarin	19834
21	19811
stadsraad	19789
prof.	19757
beleggers	19756
moord	19736
km	19722
werkers	19687
juis	19616
voorste	19605
hand	19586
ooreenkoms	19583
soort	19554

hulp	19513
senior	19508
gewys	19496
spesiale	19427
bewering	19419
herstel	19357
Franse	19300
vertoning	19299
partye	19286
toernooi	19266
Maandag	19249
eens	19239
langs	19232
veld	19229
heelwat	19213
Paul	19189
res	19177
verdere	19172
hoewel	19145
talle	19055
afdeling	19030
/	19025
dorp	19020
Noord-Transvaal	19019
Ander	18995
veroorsaak	18986
optrede	18981
graag	18973
Provinsie	18948
17	18871
ouers	18868
paaltjies	18852
boek	18846
uiteindelik	18846
Party	18816
Mense	18776
bewys	18751
omgewing	18750
verwagting	18715
proses	18702
nuus	18662
verdien	18651
woon	18648
rugby	18617
aansoek	18565
Sondag	18546
heeltemal	18521

rand	18507
200	18422
skerp	18417
stadium	18401
bal	18362
oorweeg	18357
leier	18354
voorval	18342
wedren	18294
ver	18291
verteenwoordig	18217
22	18216
Park	18181
bank	18171
kennis	18152
dis	18139
beplan	18108
dollar	18106
kaptein	18062
val	18022
aangestel	17944
toer	17930
KAAPSTAD	17862
kant	17767
Springbokke	17764
Peter	17751
plan	17736
tydperk	17717
Europese	17712
Roux	17711
Baie	17704
gou	17672
26	17654
poging	17630
moontlike	17629
daarvoor	17616
Venter	17605
vroeg	17595
geskiet	17585
gebiede	17532
bekende	17451
leer	17441
gisteroggend	17423
inkomste	17413
middel	17332
Haar	17329
geraak	17294

bevestig	17291
19	17285
ernstige	17260
betref	17253
My	17232
voltooi	17214
provinsiale	17209
boekjaar	17204
kg	17188
Bank	17187
trek	17185
sowel	17113
gelewer	17072
500	17049
erken	17018
oop	16985
wins	16983
waarom	16972
motors	16945
hospitaal	16897
beseer	16877
geskryf	16861
vinnig	16852
verwys	16848
aksie	16827
Plessis	16796
aangewys	16770
posisie	16761
enkele	16749
gistermiddag	16743
23	16742
aantal	16706
George	16703
27	16640
gelei	16615
bestuurder	16596
watter	16588
totale	16586
voer	16498
projek	16488
binnekort	16482
By	16469
produkte	16454
soveel	16424
hande	16417
moeilik	16398
Nelson	16374

Tot	16351
huise	16331
aangeteken	16311
uitgereik	16310
vierde	16308
beleid	16301
Dan	16286
organisasie	16274
Sapa-AP	16257
geplaas	16207
tog	16161
verminder	16116
Zimbabwe	16114
Frankryk	16112
nege	16105
belangrik	16031
stap	16028
bo	16012
rolprent	15991
soms	15981
Groot	15918
hoor	15899
Meer	15889
kantoor	15881
Kruger	15879
gelê	15878
gewone	15862
Pieter	15806
geklop	15766
Uit	15759
toon	15737
eintlik	15725
Gauteng	15692
geniet	15678
gewoonlik	15659
gedruk	15618
Fourie	15609
Hierdie	15540
31	15534
taal	15521
gestuur	15520
Jy	15494
Vandag	15423
Rensburg	15415
stelsel	15407
Japan	15366
Europa	15327

doel	15297
toestand	15293
Swart	15276
28	15231
James	15229
geskiedenis	15196
haal	15188
daaraan	15183
Meyer	15172
leerlinge	15168
bou	15134
minstens	15133
gekies	15123
styging	15094
wyse	15083
Mens	15075
provinsie	15059
bedrywighede	15037
Du	15032
Piet	14984
bietjie	14971
algemene	14941
Indien	14939
waarmee	14930
werklik	14917
Wyk	14894
stuur	14840
gevaar	14793
ry	14780
voorheen	14779
vertoon	14769
50	14761
Sowat	14717
All	14686
R1	14683
daaroor	14679
besturende	14663
mooi	14637
eenheid	14616
012	14595
dusver	14595
Soos	14580
verdienste	14562
Nel	14520
Man	14506
Wet	14499
kommissie	14494

OP	14475
speler	14465
Pres.	14463
bestaande	14436
lui	14435
Louis	14433
moontlikheid	14420
verskil	14412
aand	14346
situasie	14346
onderneming	14314
Andrew	14312
verkry	14267
verklaar	14246
getref	14224
genoem	14220
beweer	14212
Duitsland	14208
twaalf	14204
Mbeki	14191
vertrek	14185
Hoe	14177
oud	14174
Zyl	14170
Nie	14136
Coetzee	14081
Departement	14074
dien	14073
omstreeks	14061
voorsien	14055
Pretorius	14050
Drie	14033
publiek	14014
ontstaan	14004
hoofsaaklik	13996
gekoop	13951
drieë	13918
verantwoordelik	13893
Elizabeth	13891
Duitse	13875
kerk	13841
gemiddelde	13824
vervang	13813
Nieu-Seeland	13783
verby	13759
byeenkoms	13722
blyk	13714

markte	13713
aanvanklik	13707
gewerk	13679
Transvaalse	13656
plaasvind	13649
omstandighede	13618
Sê	13612
belangrikste	13566
reaksie	13560
sukses	13556
Martin	13532
29	13528
plekke	13523
voort	13514
versoek	13502
los	13486
Richard	13468
leiers	13456
Ook	13445
ernstig	13437
oë	13422
opleiding	13416
woorde	13393
Wanneer	13357
sommer	13336
Villiers	13333
styg	13328
ontken	13326
amptelike	13316
ding	13316
Strydom	13315
Mike	13290
wenner	13287
aanval	13276
400	13242
meter	13223
Donderdag	13214
ondanks	13197
God	13194
bydrae	13175
hoog	13171
lees	13167
Oor	13159
rentekoerse	13140
verlaat	13123
Toit	13121
homself	13072

verhouding	13068
onderwys	12993
Me.	12977
krag	12974
brief	12956
langer	12944
gepraat	12911
daarmee	12903
lekker	12886
optree	12864
sektor	12857
gevalle	12851
Engels	12816
ken	12797
keuse	12792
samewerking	12784
Viljoen	12775
gelyk	12771
kwartaal	12767
vanaand	12756
groepe	12751