Bachelorproject
Multi-agent Model van Taalverandering

Gert van Valkenhoef
g.h.m.van.valkenhoef@ai.rug.nl

Begeleider: Bart de Boer

Rijks*universiteit* Groningen

5 maart 2007

Samenvatting

Er wordt onderzocht hoe een multi-agent model taalverandering kan verklaren. Daartoe wordt taaldiversiteit geoperationaliseerd aan de hand van een aantal kwalitatieve fenomenen. Het voorgestelde model wordt aan een aantal experimenten onderworpen, die de verschillende componenten zoveel mogelijk individueel beschouwen. Er worden met het model als geheel experimenten gedaan die gestaafd worden aan de hand van de operationalisatie van taaldiversiteit. Van de mechanismen waaruit het model opgebouwd is, wordt beargumenteerd waarom deze realistisch zijn. Daarnaast is de analyse van dergelijke modellen onderwerp van discussie en wordt een poging gedaan dit op te lossen door middel van een clusteranalyse.

1 Inleiding

Multi-agent modellen stellen ons in staat om te zien hoe complexe sociale fenomenen voort kunnen komen uit eenvoudige gedragingen van individuele agents. De kracht van deze methode is, dat vaak blijkt dat een verschijnsel op een veel eenvoudigere manier verklaard kan worden dan op het eerste gezicht het geval lijkt. Ook op het gebied van taal zijn dergelijke modellen al succesvol ingezet (Axelrod, 1997; Barr, 2004; Buzing et al., 2005; Nettle, 1999; Wang & Minett, 2005).

Vaak gebruiken deze modellen een zeer eenvoudige abstractie van taal, in een simulatie waar de agents als doel hebben met elkaar te communiceren. Deze modellen laten zien dat er geen algemene autoriteit of common knowledge nodig is om te verklaren dat agents dezelfde taal leren: het adaptief gebruik van de taal op zich is voldoende (Barr, 2004). In een model waar cultuur en communicatie elkaar wederzijds beïnvloeden (Axelrod, 1997), blijken een klein aantal grote gebieden te ontstaan die dezelfde taal en cultuur aannemen.

Waar deze modellen echter geen verklaring voor bieden, is het voortdurend veranderende karakter van taal en het ontstaan van een 'gradiënt' van dialectsprekers (waarbij elk dorp een soort mengvorm van de dialecten in naburige dorpen spreekt).

Dit onderzoek zal een exploratief onderzoek zijn, waarbij uitgegaan wordt van een vrij eenvoudig model voor taalverandering. Belangrijkste uitgangspunt is het model van Axelrod (Axelrod, 1997). Op dit model worden een aantal uitbreidingen gedaan, om te kijken naar de invloed van bepaalde mechanismen op het taalveranderingsproces. Daarom zullen de resultaten vooral kwalitatief van aard zijn.

De volgende aannames dienen om het onderzoeksgebied in te perken:

- Het taalvermogen van de agents is gelijk. Het doel is immers om taalverandering te onderzoeken in een populatie die al talig is.
- Het model zal niet evolutionair zijn, waarbij kinderen de taal van hun ouders overnemen, maar

een dynamischer model waarbij de agents taal van elkaar leren.

- Een taal biedt geen inherent voordeel boven andere talen. Er is geen sprake van functionele bias, Nettle (Nettle, 1999) doet hier bijvoorbeeld al onderzoek naar.
- Communicatie wordt als een doel op zich beschouwd, agents hebben geen externe motivatie die de communicatie intentie geeft. Een interessant SugarScape model doet een eenvoudige poging om dit wel te doen (Buzing et al., 2005).
- Het model is geografisch van aard, omdat dit een belangrijk aspect van de echte wereld is en hier in het verleden al hoopgevende resultaten mee behaald zijn (Axelrod, 1997; Barr, 2004; Buzing et al., 2005).

Het model zal (gegeven verschillende beginsituaties) verschillende fenomenen moeten kunnen verklaren, namelijk zowel het ontstaan van diversiteit als het toenemen van homogeniteit in de populatie.

Uit onderzoek naar chaotische systemen is al gebleken dat vrij eenvoudige systemen kunnen leiden tot een complexe diversiteit, zelfs gegeven een vrijwel homogene beginsituatie (Pearson, 1993). Dat is een sterke indicatie dat ook een (eenvoudig) multi-agent systeem het ontstaan van een divers taalgebied moet kunnen verklaren.

1.1 Onderzoeksvraag

Gegeven een basaal grid-based multi-agent model, waarin op een abstracte manier taal gemodelleerd is, welke mechanismen zijn nodig om het ontstaan en behoud van taaldiversiteit, zoals we dat in de echte wereld tegenkomen, te verklaren?

Daarbij hoort natuurlijk wel de voorwaarde, dat voor de gepostuleerde mechanismen een plausibel evenbeeld gevonden kan worden in de werkelijkheid van menselijke samenleving.

Het te meten construct is, kort door de bocht, taaldiversiteit. Natuurlijk is dit niet gemakkelijk kwantitatief te meten of te visualiseren. Maar, dit is een exploratief onderzoek naar de benodigde mechanismen om (dynamische) taaldiversiteit te verklaren. Daarom is het nuttig om het begrip 'dynamische taaldiversiteit' in termen van een viertal kwalitatieve fenomenen te operationaliseren.

- Ten eerste convergeert het taallandschap nooit geheel naar één taal, maar is er altijd sprake van het ontstaan en verdwijnen van verschillende dialecten.
- Ten tweede mag het taallandschap ook niet geheel divergeren: eenlingen die een andere taal spreken dan hun omgeving, zullen zich onder druk van die omgeving aanpassen en een meer gangbare taal aannemen.
- Ten derde, in ieder geval bij een beginsituatie met twee zeer verschillende talen een scherpe taalgrens waarover de twee talen strijden. Dus wel een duidelijke grens, maar een grens die zich wel voortdurend kan verplaatsen.
- Ten vierde zal het model ook dialect-gradiënten moeten laten zien, waarbij elke opvolgende agent een iets ander dialect spreekt.

Naast het feit dat het model de dynamische taaldiversiteit moet kunnen verklaren, moet het dat op een aannemelijke manier doen. Dat wil zeggen: de gebruikte mechanismen moeten gebaseerd zijn op, of parallellen hebben met mechanismen in de realiteit.

2 Methode

Voor de simulaties is gebruik gemaakt van een multi-agent simulatie-omgeving, "The Recursive Porous Agent Simulation Toolkit" (Repast) (North et al., 2006). Deze omgeving is gekozen naar aanleiding van een review (Railsback et al., 2006) waarin verschillende omgevingen vergeleken worden. Doorslaggevende punten waren de implementatie in Java (veel omgevingen zijn geschreven in Objective-C, waar ik geen ervaring mee heb), de relatief goede prestaties en vrij uitgebreide class library. Hoewel de volledigheid en betrouwbaarheid van deze classes vaak te wensen over laat, ben ik redelijk tevreden over de keuze voor Repast.

Hoewel deze toolkit dus de nodige problemen kent, zijn die niet onoverkomelijk. Soms was het nodig een andere methode te kiezen, maar het feit dat de broncode beschikbaar is zorgt er ook voor dat fouten of tekortkomingen vaak vrij snel aangepakt kunnen worden. Hoewel dit natuurlijk geen gewenste activiteiten zijn, heeft het kant en klaar beschikbaar zijn van een toch vrij uitgebreide omgeving de implementatie wel gemakkelijker gemaakt.

De gebruikte analysegereedschappen (zie sectie 2.2) omvatten wat globale statistieken, een visualisatie van lokale verschillen en een globale clustering

Parameter	Symbool	Default
Wereldgrootte x-richting	S_x	10
Wereldgrootte y-richting	S_y	10
Aantal features	f	15
Aantal traits	t	5
Neighborhood straal	r	2
Groepgrenswaarde	$\hat{\tau}$	0.3
Mutatiekans	p_m	0.01
Translatie paniekcurve	δ	0.2
Stijlheid paniekcurve	γ	38
Conformance	λ	1
Stijlheid convergentiecurve	α	3
Stijlheid divergentiecurve	β	5

Tabel 2.1: Lijst van modelparameters

om taalgebieden van elkaar te kunnen onderscheiden. De implementatie komt uitgebreid aan bod, om reproductie mogelijk te maken.

De uitgevoerde analyses worden ook, terwijl het model draait, voortdurend in beeld gebracht (hoewel het mogelijk is om dit in Repast uit te schakelen). Voor een bespreking van de gebruikte technieken, zie sectie 2.3.

2.1 Model

Het model is in feite een complexe variant van het model van Axelrod (Axelrod, 1997). Het is een spatieel model, waarbij de agents gedragingen vertonen in de context van hun nabije omgeving (neighborhood). Die neighborhood is aanzienlijk groter dan in het model van Axelrod. De agenten beschikken ieder over een taal en zullen er naar neigen dezelfde taal te spreken als hun omgeving. Daarnaast laten agents zich niet zomaar uitzonderen (bij het model van Axelrod ontstaan vaak taalgebieden bestaande uit één of twee agents) en is er een 'novelty drive', indien hun omgeving vrij homogeen is voelen agents zich juist aangetrokken door uitzonderlijke eigenschappen.

Voor een lijst van alle parameters, zie tabel 2.1. De parameters worden toegelicht in de hieropvolgende bespreking van het model.

2.1.1 Representatie

Het model is discreet spatieel georganiseerd, de wereld waarin de agents leven bestaat uit een $m \times n$ raster. Dit raster is toroïdaal, wat betekent de extremen in zowel x als y richting met elkaar in verbinding staan, waardoor een torusvorm (donut-) ontstaat. In elk vakje van dit raster leeft één agent, die dus *altijd* 4 directe buren heeft (noord, oost, zuid, west).

Elke agent heeft één taal, welke gerepresenteerd is als een vector van f 'features', waarbij elke feature t mogelijke waarden, of 'traits' heeft. De terminologie is analoog Axelrod (1997).

De afstand tussen twee talen is gedefineerd als de Euclidische afstand tussen de verschillende taalvectoren (zie formule 2.1). Om de afstand op deze manier te defineren heeft onder andere als voordeel dat clustering (en andere handelingen waarvoor een afstand van belang is) makkelijker is toe te passen.

$$d(x, y) = \sqrt{\sum_{i=1}^{f} (x_i - y_i)^2} \tag{2.1}$$

2.1.2 Neighborhood

De agent ziet zichzelf in de context van andere nabije agents, zijn neighborhood. De neighborhood is vrij groot omdat de agent binnen die groep agents groepen moet kunnen onderscheiden. Dit zijn de culturele groepen waar de agent wel of niet onderdeel van is. Op basis van een aantal eigenschappen van die groepen (grootte van de groep, variatie binnen de groep) zal de agent bepaalde gedragingen vertonen.

De gekozen neighborhood is een Moore neighborhood (Weisstein, 2003a), de keuze hiervoor is met name geïnspireerd door de defecte implementatie van de Von Neumann neighborhood (Weisstein, 2003b) in Repast. De Moore neighborhood is een vierkante neighborhood met een straal r. Het aantal neighbors is dan

$$N = (2 \times r + 1)^2$$

De standaardinstelling is $r = 2$, wat neer komt op een neighborhood size $N = 25$.

2.1.3 Groepen

Om de culturele groepen te onderscheiden wordt op deze neighborhood een complete-link clustering uitgevoerd (Jain et al., 1999; King, 1967). Dit is een hiërarchisch clustering algoritme dat begint met elk datapunt in zijn eigen cluster. Daarna worden steeds de clusters met minimale onderlinge afstand samengevoegd. De afstand tussen twee clusters is gedefinieerd als het maximum van de onderlinge afstanden van de datapunten in de twee clusters. Dit resulteert in compacte clusters. Elke samenvoeging van clusters wordt geannoteerd met deze afstand. De clustering

is te beschouwen als een graaf die de datapunten verbindt.

Er is een grenswaarde, τ, die bepaalt welke agents de agent beschouwt als onderdeel van zijn groep. Alle bogen met een lengte groter dan τ, worden verwijderd, waardoor een aantal clusters met een straal kleiner dan τ overblijft.

Deze grenswaarde is overigens niet een directe parameter van het model. De waarde zo immers niet erg robuust zijn ten opzichte van het aantal features of traits. Daarom is er voor gekozen hier een meer robuuste parameter te nemen, die aan de hand van f en t de waarde voor τ berekent. Dit is een factor $\hat{\tau}$ tussen 0 en 1, die vermenigvuldigd wordt met de maximale afstand om τ te krijgen (zie formule 2.2).

$$\tau = \hat{\tau} \times \sqrt{f \times t^2} \tag{2.2}$$

Nu beschouwen we het cluster waar de agent zelf onderdeel van is. Dit cluster heeft bepaalde grootte, het aantal leden n van het cluster. Dit getal relatief aan de grootte van de neighborhood N geeft de 'comfort' c (zie formule 2.3), een maat voor in hoeverre de agent zich in zijn omgeving op zijn gemak voelt. Daarnaast levert de maximale afstand tussen punten in het cluster, $d_{\max}$ ten opzichte van de grenswaarde τ een maat voor variatie binnen het eigen cluster, de 'cluster spread' s (zie formule 2.4), ofwel 'spread'.

$$c = \frac{n}{N} \tag{2.3}$$

$$s = \frac{d_{\max}}{\tau} \tag{2.4}$$

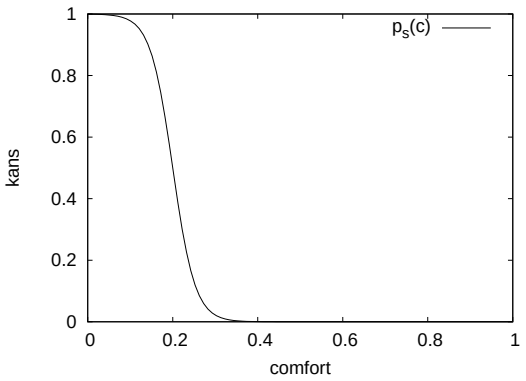
Van deze twee maten, beide waarden tussen 0 en 1, kan een derde afgeleid worden. Deze is van centraal belang voor de gedragingen van de agent. Dit is een maat van verveling ('boredom') b (zie formule 2.5). De boredom is hoog als de comfort hoog is (de agent is erg op zijn gemak) en de spread laag is (er zijn weinig 'spannende' gesprekspartners).

$$b = c \times (1 - s) \tag{2.5}$$

2.1.4 Interacties

Het gedrag van de agent bestaat uit een viertal mogelijke gedragingen:

- Mutatie
- Paniek
- Convergeren



Figuur 2.1: p_s, de kans dat een agent in paniek raakt ('switch' gedrag gaat vertonen)

- Divergeren

Of één van deze gedragingen uitgevoerd wordt, is voor elke gedraging afhankelijk van een kansfunctie. Voordat deze kansfunctie geëvalueerd wordt, wordt eerst volgens een uniforme verdeling willekeurig één van de vier mogelijke gedragingen gekozen. Voor deze gedraging wordt dan de betreffende kansfunctie gebruikt om te beslissen of de gedraging uitgevoerd mag worden.

De eerste gedraging, mutatie, gebeurt met een vaste kans p_m. Hierbij krijgt één willekeurige feature een willekeurige trait.

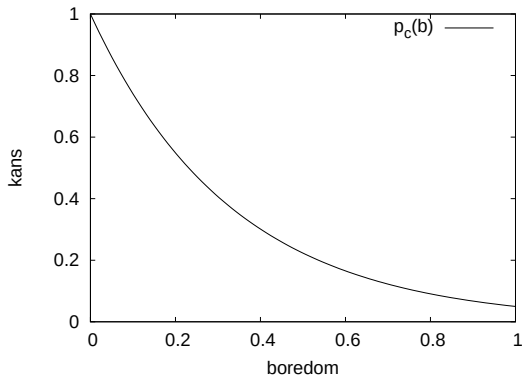
De kans p_s dat de agent in paniek raakt is afhankelijk van de comfort als sigmoïde functie (zie formule 2.6). Die heeft twee parameters, een translatie δ en een factor γ die de stijlheid van de curve bepaalt.

$$p_s = 1 - \frac{1}{1 + e^{-\gamma(c-\delta)}} \tag{2.6}$$

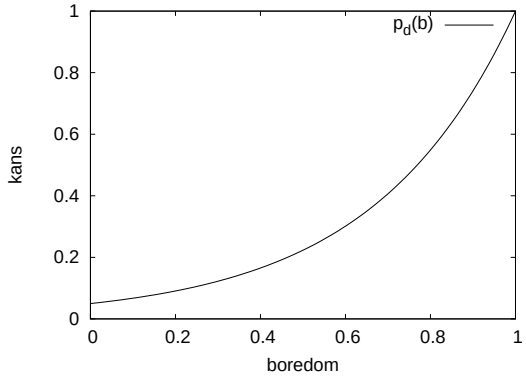
Raakt de agent in paniek, dan zal hij aan de hand van de kansdistributie R (zie sectie 2.1.6) een ander cluster kiezen (het eigen cluster wordt hierbij niet meegenomen), en daaruit een willekeurige agent kiezen (volgens een uniforme kansverdeling). Door de eigenschappen van R hebben grote clusters een grotere kans om gekozen te worden. Aan de hand van de 'conformance' parameter λ ($0 \leq \lambda \leq 1$) wordt vervolgens de gehele taal van de agent herzien (zie formule 2.7).

$$L_{\text{new}} = \lambda \times L_{other} + (1 - \lambda) \times L_{current} \tag{2.7}$$

De agent kan aan de hand van de 'boredom' functie convergent gedrag, divergent gedrag of geen van beide



Figuur 2.2: p_c**, de kans dat een agent convergent gedrag gaat vertonen.**



Figuur 2.3: p_d**, de kans dat een agent divergent gedrag gaat vertonen.**

gaan vertonen. Deze gedragingen hebben betrekking op de plaats van de agent binnen zijn cluster. Informatie over omringende clusters wordt hiervoor dus niet meer gebruikt.

Zowel de convergentie (p_c) als divergentie (p_d) kans zijn exponentiële functies van de boredom (zie formule 2.8, 2.9). Bij beide gedragingen ordent de agent de andere agents in zijn cluster naar de taalafstand tussen zijn eigen taal en de taal van de andere agents. Hier uit wordt een agent gekozen aan de hand van de distributie R (zie sectie 2.1.6). Voor het convergente gedrag hebben gelijkende agents een hogere kans gekozen te worden, voor het divergente gedrag de minder gelijkende agents.

$$p_c = e^{-\alpha b} \tag{2.8}$$

$$p_d = e^{-\beta(1-x)} \tag{2.9}$$

Het convergente gedrag houdt in, dat de agent een willekeurige feature uit zal kiezen en hiervoor de trait van de gekozen partner over zal nemen.

Het divergente gedrag is wat complexer, hier is namelijk sprake van de 'novelty drive': de agent zoekt een attribuut uit die door hem 'spannend' wordt bevonden. Er zijn verschillende manieren om te bepalen wat een agent spannend vindt.

Methode 1 Per feature wordt de variantie binnen het cluster en de absolute afstand voor deze features tussen de agents genomen. Vervolgens worden beide sets waarden genormaliseerd aan de hand van de maximaal gevonden waarde.

De agent zal voorkeur geven aan een feature met lage variantie binnen het cluster (bijna iedere agent heeft hiervoor dezelfde waarde) maar een groot verschil in traits met de gekozen partner. Hiervoor wordt een 'appeal' waarde a_i (zie formule 2.10) voor elke feature i berekend en een ordening naar deze waarde gemaakt. Aan de hand van de distributie R (zie sectie 2.1.6) kiest de agent een feature en wisselt de waarde hiervan uit met de gekozen partner.

$$a_i = (1 - \frac{Var(X_i)}{\max_j Var(X_j)}) \times \frac{d_i}{d_{max}} \tag{2.10}$$

$Var(X_i)$ is de variantie op feature i, en feature-afstand $d_i = |x_i - y_i|$, waar x de taalvector van de agent en y de taalvector van de gekozen partner. De maximale feature-afstand is gegeven door $d_{max} = \max_i d_i$.

Methode 2 De agent berekent voor elke feature van de gekozen partner hoeveel andere agents (zichzelf inbegrepen) dezelfde trait hebben. Dit ten opzichte van de grootte van het eigen cluster, levert een maat voor appeal (zie formule 2.11). Bij deze waarde kunnen we heel eenvoudig een grenswaarde vaststellen: het is natuurlijk niet realistisch om een trait die door de helft van de agents gedeeld wordt te beschouwen als 'novel'. Dus de appeal zal moeten voldoen aan $a_i > \frac{1}{2}$.

$$a_i = (1 - \frac{n_{same}}{N}) \tag{2.11}$$

Uit alle attributen die overblijven na selectie aan de hand van a_i wordt met behulp van de distributie R (zie sectie 2.1.6) een feature gekozen. De agent neemt de trait voor deze feature over van zijn partner.

2.1.5 Uitvoering

Bij uitvoering van het model, wordt eerst het raster aangemaakt waarop de agents leven. Vervolgens worden de agents hierin geplaatst (met òf een vooraf gespecificeerde prototype taal, òf een willekeurig geïnitialiseerde taal). Zodra alle agents aangemaakt zijn, wordt voor iedere agent de neighborhood bepaald. Deze neighborhood zal hetzelfde blijven gedurende de uitvoering van het model (hoewel de agents in de neighborhood zelf wel kunnen veranderen).

Voor elke uitvoeringsstap wordt de lijst agents willekeurig herordend, zodat de agents in willekeurige volgorde aangesproken worden. Dit om enige invloed van de ordening van de lijst agents uit te sluiten. Elke agent voert zijn stapfunctie, de gedragingen zoals beschreven (zie sectie 2.1.4), uit.

Nadat de gedragingen uitgevoerd zijn, komt de analyse en visualisatiestap. De bewerkingen die in deze stap worden uitgevoerd, worden beschreven in de sectie over meetmethoden (zie sectie 2.2).

2.1.6 Getransformeerde uniforme verdeling

Op een aantal plaatsen in het model wordt gebruik gemaakt van een verdeling die de uniforme kansverdeling zo transformeert, dat de indices van laag naar hoog een lineair oplopende kans hebben. Hier wordt de gebruikte transformatie toegelicht. Deze toelichting is natuurlijk ook van toepassing op het geval dat de lage indices juist een hogere kans hebben, dit wordt bereikt door de opgeleverde index te transformeren.

We willen een willekeurig getal $i \in [1, n]$, waarbij de hoogste index n de hoogste kans heeft om gekozen te worden, lineair aflopend tot de laagste kans voor index 1. Beschouw nu de verdeling

$$R(1, n) = [\sqrt{U(0, n^2 - 1)}] + 1 \tag{2.12}$$

waarbij U de uniforme distributie is en $[x]$ de grootste integer is z.d.d. $[x] \leq x$.

Dat $R(1, n)$ inderdaad een getal oplevert in $[1, n]$, zien we door de laagst en hoogst mogelijke waarden voor de uniforme willekeurige waarde te nemen, bij $U = 0$ krijgen we $R(1, n) = 1$ en bij $U = n^2 - 1$ krijgen we n. Zou het bereik van de uniforme variabele groter zijn dan zou de wortel imaginaire waarden opleveren, of $R(1, n) > n$

Voor de kans dat $R(1, n)$ een bepaalde waarde i oplevert is gegeven door $P(R = i) = P((i-1)^2 \leq [\sqrt{}(0, n^2 - 1)] < i^2) = \frac{i^2 - (i-1)^2}{n^2} = \frac{2i+1}{n^2}$, hetgeen inderdaad een lineair oplopende kans met de index i is.

2.2 Meetmethoden

De entropieberekeningen worden uitgevoerd en geplot (zie sectie 2.2.1), elke agent rekent de afstand tot zijn directe buren uit (zie sectie 2.2.2) en de gehele lijst agents wordt geclusterd (zie sectie 2.2.3).

2.2.1 Entropie

Entropie is een maat voor de hoeveelheid onzekerheid die er over een variabele is. Hoe meer onzekerheid er is, hoe meer nieuwe informatie het weten van een waarde heeft. De entropie is dus maximaal wanneer de data volledig willekeurig verdeeld is en minimaal als de variabele altijd dezelfde waarde heeft. De entropie is gedefinieerd in termen van de kans op elke gebeurtenis waarvan de uitkomst bekend is (zie formule 2.13) (Weisstein, 2007).

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{2.13}$$

De kansen kunnen afgeleid worden uit de huidige verdeling binnen de populatie. Op die manier wordt per feature de entropie berekend, door per trait het aantal agents te tellen dat die trait heeft. Dat geeft formule 2.14, waarbij n_i het aantal agents met trait i is en N het totaal aantal agents.

$$H(X) = -\sum_{i=1}^{t} \frac{n_i}{N} \log_2 \frac{n_i}{N} \tag{2.14}$$

De gemiddelde, minimale en maximale entropie geven vervolgens een maat voor de hoeveelheid variatie binnen het systeem.

2.2.2 Lokale afstanden

De afstanden van een agent tot zijn directe buren kan een indicatie zijn voor het bestaan van een taalgradiënt, of juist een scherpe taalgrens. Deze kan echter niet volledig vertrouwd worden, mede omdat de neighborhood van een agent meer beslaat dan slechts zijn directe buren.

2.2.3 Globale clustering

De methode voor globale clustering is een KMeans clustering in combinatie met een methode die poogt het juiste aantal clusters en goede seed points te

schatten (Jain et al., 1999). Het algoritme heet ISODATA (Ball & Hall, 1965) en gebruikt in de originele vorm een aantal heuristieken gebaseerd op grenswaarden. Omdat het niet erg handig is om voor elke set parameters die gebruikt worden ook een aantal parameters voor de clustering te moeten bepalen, wordt voor de globale clustering een variant gebruikt die de originele heuristieken vervangt door heuristieken uit de informatietheorie, hetgeen een bijna parametervrije clustering oplevert (Carman & Merickel, 1990).

Voor een lijst van parameters, zie tabel 2.2. De gebruikte criteria waaraan in het volgende verwezen worden zijn samengevat in tabel 2.3

Centraal in het gebruikte algoritme, CAIC ISODATA (Carman & Merickel, 1990), is het CAIC (Consistent Akaike Information Criterion). Deze is een variant op het AIC (Akaike Information Criterion) die wel asymptotisch consistent is (de variantie van het CAIC gaat naar nul, als het aantal datapunten naar oneindig nadert). Een lage waarde voor dit criterium betekent een goede fit van het model aan de data, maar overfitting wordt bestraft (zie formule 2.15).

$$C = np(1 + \log_e(2\pi)) + \sum_{j=1}^{k} n_j \log_e |S_j| + 2kp \log_e(n) \tag{2.15}$$

Waarbij n het aantal datapunten, p het aantal dimensies in de dataset, k het aantal clusters in het model, n_j het aantal datapunten in cluster j en $|S_j|$ de determinant van de (geschatte) variantie-covariantiematrix voor cluster j.

CAIC ISODATA zoekt naar een minimum van de waarde voor C. Daarvoor wordt een globale C_{min}, de minimaal gevonden C bijgehouden en voor elke hoeveelheid clusters die al geprobeerd is een waarde C_k, de minimum C bij k clusters. Bij de minimale C_{min} hoort een clustering die ook opgeslagen wordt (de voorlopig beste clustering, c_{best}).

De uitvoering van het algoritme begint met alle datapunten in één cluster. De hieraan verbonden CAIC waarde wordt berekend.

Alle clusters met minder dan twee datapunten worden weggegooid (thin). In het originele artikel wordt een thinning threshold van 1 gebruikt (Carman & Merickel, 1990), maar omdat (om redenen die later duidelijk zullen worden) het nodig is om de varianties van de variantie-covariantiematrix te berekenen, zijn er minstens drie datapunten nodig.

Vervolgens worden aan de hand van een aantal criteria de clusters gespleten (split) of samengevoegd (merge).

Algoritme 2.1 CAIC ISODATA

```
c ← initialClustering()
repeat
  c ← kMeans(c)
  c ← thin(c)
  k ← size(c)
  C ← CAIC(c)
  if unknown(C_k) or C < C_k then
    C_k ← C
  end if
  if unknown(C_min) or C < C_min then
    C_min ← C
    c_best ← c
  end if
  if Splitting criteria are met then
    c ← split(c)
  else if Merging criteria are met then
    c ← merge(c)
  end if
until Stopping criteria are met
```

Bij de split operatie wordt het cluster met de grootste variantie ($|S_j|$) gekozen. Langs de as met de grootste standaarddeviatie σ_{max} worden de twee nieuwe centroïden (clustercentra) in tegenovergestelde richting een percentage van σ_{max} verplaatst.

Bij de merge operatie worden de datapunten van beide clusters in één cluster geplaatst en wordt hiervan de centroïde berekend.

Na het splitten of mergen wordt een KMeans clustering (Jain et al., 1999) uitgevoerd met de bepaalde (nieuwe) seed points. Hieraan is een maximaal aantal iteraties KMeans verbonden. Hierna wordt weer de CAIC waarde uitgerekend en de cyclus begint opnieuw.

Het algoritme stopt als het maximale aantal iteraties bereikt is, of het tot de conclusie komt dat het niet waarschijnlijk is dat er een betere clustering gevonden zal worden. Voor de parameters en criteria, zie tabel 2.2 en 2.3. Het algoritme wordt nog eens samengevat in algoritme 2.1.

Een heikel punt bij het gebruik van dit algoritme is de determinant $|S_j|$ van de variantie-covariantiematrix van elk cluster. De variantie-covariantiematrix is vrij eenvoudig te schatten. De empirische variantie-covariantiematrix S kan lineair in het aantal datapunten n en kwadratisch in het aan-

Parameter	Symbool	Default
Maximum aantal ISODATA iteraties	I	20
Maximum aantal KMeans iteraties	I_k	20
Doelaantal clusters	K	10
Maximum verandering in aantal clusters per iteratie	N	1
Fractie van de standaarddeviatie om van het gemiddelde te verwijderen (bij splitten)	α	0.25
Maximum aantal datapunten waarbij thinning toegepast wordt	Θ_N	2

Tabel 2.2: CAIC ISODATA parameters

Name	Rule
Attempt splitting if	$C < C_{k-1}$
Split cluster if	$\|S_j\| > \frac{1}{k}\sum_{j=1}^{k}\|S_j\|$ **or** $k \leq \frac{K}{2}$
Attempt merging if	$C > C_{k-1}$
Merge cluster pair if	D_{ij} is the smallest distance, $i \neq j$
Stop if	iter $= I$ **or** ($C > C_k$ **and** $C - C_k < 0.01$ **and** $C_k \neq C_{min}$) **or** ($C > C_{min}$ **and** $C - C_{min} < 0.001$)

Tabel 2.3: CAIC ISODATA criteria

tal dimensies p berekend worden. De waarden zijn

$$s_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j) \qquad (2.16)$$

waar $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}$ en x_{ki} observatie k van variabele X_i (Schäfer & Strimmer, 2005).

Echter, zoals Schäfer & Strimmer (2005) aangeven, hebben zowel de empirische variantie-covariantiematrix als de maximum likelihood estimate $S^{ML} = \frac{n-1}{n}S$ behoorlijke problemen met kleine n, grote p problemen. S is dan geen goede schatting van de werkelijke matrix Σ en bovendien krijgt de determinant geen mooie waarden (de determinant moet een waarde groter dan 0 krijgen voor het berekenen van de CAIC (zie formule 2.15)). Schäfer & Strimmer (2005) geeft een methode die in ongeveer twee keer de tijd die nodig is om S te berekenen een betere matrix $S^\star$ oplevert die deze vervelende eigenschappen niet heeft. Deze methode heet 'shrinkage', voor de exacte berekening verwijs ik naar Table 1, Schäfer & Strimmer (2005, p. 4) en voor de volledigheid vermeld ik nog dat de 'shrinkage target' D gekozen is.

In de berekening van $S^\star$ worden ook de varianties van de variantie-covariantiematrix gebruikt. Om dat op een zinnige manier te doen zijn tenminste drie datapunten nodig (bij twee datapunten is door invullen eenvoudig te zien dat anders de waarden 0 worden, ongeacht de data).

Een tweede probleem is dat de determinant van de covariantiematrix 0 wordt indien één van de variabelen geconvergeerd is naar eenzelfde waarde voor alle agents in een cluster. De oplossing hier is om wat normaal verdeelde 'ruis' bij de waarden op te tellen, zodat de variantie nooit naar 0 gaat.

De clustering wordt niet rechtstreeks op de data uitgevoerd, maar elke agent krijgt een ruismasker mee die bij de werkelijke waarden opgeteld wordt. Deze ruis is normaal verdeeld met een $\mu = 0$ en standaarddeviatie σ. De waarde van σ zal later nog aan bod komen (zie sectie 3). Een nadeel is natuurlijk wel dat de interpretatie van de clustering complexer is omdat de clusters ook een artefact kunnen zijn van deze ruisverdeling.

2.3 Visualisatie

Het model wordt op een aantal manieren 'real-time' gevisualiseerd:

- Een twee-dimensionale weergave van de rasterwereld
- Een dynamisch staafdiagram waarin de per-feature entropie weergegeven is
- Een plot van de minimale, gemiddelde en maximale entropie tegen de tijd

De laatste twee zijn eenvoudige weergaves van de eerder besproken analyses (zie sectie 2.2.1). Ik zal hier niet verder over uitwijden.

Over de eerste visualisatie valt meer te vertellen. Dit is een twee-dimensionale weergave van de rasterwereld waarin de agents zich bevinden. Er worden verschillende soorten informatie geïntegreerd.

Het is een zwart vlak opgedeelt in een vakje per agent. De lijnen om de vakjes heen krijgen een kleur aan de hand van de afstand tussen de twee aangrenzende agents ten opzichte van de maximale afstand:

$$v = \frac{d}{\sqrt{f \times t^2}} \tag{2.17}$$

Deze visualiseer ik aan de hand van de 'black body' colormap, waarbij de kleur verloopt van zwart voor lage waarden, via rood en geel naar wit (zie formule 2.18). Deze representatie heeft als voordeel dat het vrij intuïtief te interpreteren is vanwege de normale associatie met warmte en de oplopende helderheid. Door de keuze voor zwart als achtergrondkleur is er daadwerkelijk geen zichtbare grens tussen agents met eenzelfde taalvector.

$$(r, g, b) = \begin{cases} (3v, 0, 0) & 0 \leq v < \frac{1}{3} \\ (1, 3(v - \frac{1}{3}), 0) & \frac{1}{3} \leq v < \frac{2}{3} \\ (1, 1, 3(v - \frac{2}{3})) & \frac{2}{3} \leq v \leq 1 \end{cases} \tag{2.18}$$

Daarnaast wordt de globale clustering gevisualiseerd door de vakjes in het midden een kleur te geven aan de hand van het cluster waarin ze vallen. Dit is minder triviaal dan het lijkt, omdat de clustering in een tijdstap niet overeen hoeft te komen met de clustering op een eerdere tijdstap. Zelfs als dat wel grotendeels het geval is, volgt niet vanzelf welke clusters precies op elkaar lijken. Bovendien is het vinden van goede kleuren niet erg triviaal als het aantal clusters erg varieert.

Daarom worden niet alle clusters gevisualiseerd, maar alleen de clusters die boven een minimaal aantal leden ε uitkomen tot een maximum aantal clusters υ. Als het maximum aantal clusters overschreden wordt, worden alleen de grootste clusters meegenomen in de visualisatie.

De kleuren worden gegenereerd in de HSV ruimte en voor elke index $i, 1 \leq i \leq \upsilon$ wordt de bijbehorende kleur gegeven door

$$(h, s, v) = (1, 1, \frac{i}{\upsilon}) \tag{2.19}$$

Om de toegewezen kleuren zo consistent mogelijk te houden, worden de leden van de oude clustering vergeleken met de huidige met een maat van gelijkenis, de Jaccard Index (zie formule 2.20). Dit levert een matrix van gelijkenissen op. Hieruit wordt steeds de grootste genomen om een paar van twee clusters te kiezen. Het nieuwe cluster krijgt dan dezelfde index i als het bijpassende oude cluster. Zodoende ontstaat er een vrij stabiele visualisatie (indien de clustering zelf redelijk stabiel is).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.20}$$

3 Resultaten

3.1 Analysemethoden

Om de clusteranalyse op een zinnige manier in te zetten voor analyse en visualisatie, is het noodzakelijk om de sterke en zwakke punten van de analyse te kennen. Met als doel de inzetbaarheid van de clustering te bepalen, is deze losgelaten op een aantal kunstmatige situaties, die zoveel mogelijk overeenkomen met de data zoals die in het model voorkomt.

In elk van de kunstmatige situaties is er sprake van 100 datapunten, met ieder 15 dimensies. De mogelijke waarden voor iedere dimensie zijn $1, 2, 3, 4, 5$. Daarnaast is de 'noise' parameter uit het model ook meegenomen, om de invloed hiervan te bepalen. De onderzochte situaties zijn in een tabel opgenomen (zie tabel 3.1). In de tabel heet een cluster heterogeen als op vrijwel alle dimensies de waarde willekeurig gekozen is (dit komt dus ongeveer overeen met een 'random' beginsituatie van het model) en homogeen als op elke dimensie de waarde vast ligt.

De resultaten van de evaluatie zijn ook gegeven in een tabel (zie tabel 3.2). Hierbij zijn situatie en 'noise' (σ) waarden gevarieerd en het resultaat is het gevonden aantal clusters (gemiddeld over 100 runs van de clustering, waarbij voor elke run de willekeurige waarden opnieuw gekozen zijn) en de standaarddeviatie van het gevonden aantal clusters.

Wat opvalt is, dat de clustering het goed doet als er ook werkelijk structuur te onderscheiden is, in die gevallen heeft het algoritme het meestal bij het juiste eind. Dit gaat vooral goed als er sprake is van (bijna) homogene clusters, hoewel in een vrijwel heterogene verdeling het ook nog behoorlijk goed kan gaan. Is er echter geen of weinig structuur, dan is het waarschijnlijk dat er clusters uitrollen, die eigenlijk niet bestaan. Die gevallen moeten dus op een andere manier afgevangen worden.

Een voordeel van de gekozen aanpak is wel, dat het algoritme clusters herkent in elk stadium van hun ontwikkeling - zelfs clusters die zich een klein beetje van de rest onderscheiden worden er uit gepikt en in een meer brede verdeling worden die kleine verschillen juist genegeerd.

A	Volledig homogeen (1 cluster)
B	Twee volledig homogene, maximaal gescheiden clusters
C	Twee volledig homogene clusters, gescheiden op twee dimensies
D	Volledig heterogene data (random)
E	Twee heterogene clusters, gescheiden op twee dimensies
F	Heterogeen, 3 clusters gescheiden op 1 dimensie
G	Heterogeen, 3 clusters gescheiden op 2 dimensies
H	Heterogeen, 4 clusters gescheiden op 2 dimensies
I	Heterogeen, 5 clusters gescheiden op 2 dimensies
J	Heterogeen, 5 clusters ieder geconvergeert op een verschillende dimensie
K	Zoals J, maar 2 dimensies per cluster
L	Zoals J, maar 3 dimensies per cluster
M	Homogeen, twee clusters $n \in \{90, 10\}$, gescheiden op 1 dimensie
N	Homogeen, drie clusters, ieder afwijkende waarde in verschillende dimensie
O	Homogeen, twee clusters, ieder afwijkende waarde in verschillende dimensie
P	Zoals O, met 5 clusters

Tabel 3.1: Verschillende situaties voor evaluatie clustering

situatie	clusters	noise	gevonden	afwijking
A	1	1	1.65	0.48
A	1	0.1	1.64	0.50
A	1	0.01	1.74	0.46
B	2	1	2.00	0.00
B	2	0.1	2.00	0.00
B	2	0.01	2.00	0.00
C	2	1	2.00	0.00
C	2	0.1	2.00	0.00
C	2	0.01	2.00	0.00
D	-	1	2.06	0.24
D	-	0.1	2.01	0.10
D	-	0.01	1.80	0.40
E	2	1	2.00	0.00
E	2	0.1	2.01	0.01
E	2	0.01	2.01	0.01
F	3	0.1	2.11	0.31
G	3	0.1	3.00	0.00
H	4	0.1	3.93	0.29
I	5	0.1	3.10	1.02
J	5	0.1	2.18	0.39
K	5	0.1	3.11	0.79
L	5	0.1	5.05	1.27
M	2	0.1	2.22	0.42
N	3	0.1	3.00	0.00
O	2	0.1	2.00	0.00
P	5	0.1	5.00	0.00

Tabel 3.2: Resultaten clustering evaluatie

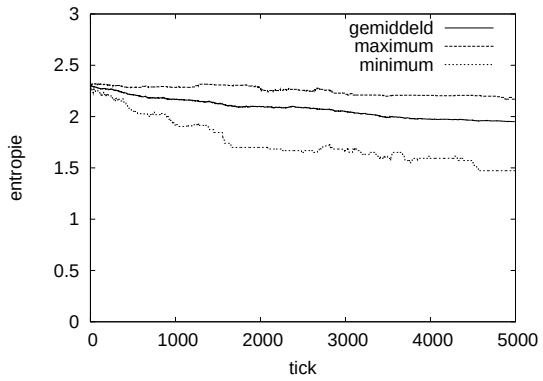
Dit is dus ideaal om, gecombineerd met een maat voor globale variatie en lokale afstanden te gebruiken om de ontwikkeling van clusters te visualiseren. Het is echter niet zo geschikt om kwantitatief conclusies te trekken over een bepaalde situatie. Hiervoor zou een thresholded clustering (zoals de originele ISODATA), uitgevoerd met verschillende thresholds, beter zijn.

Opvallend aan deze evaluatie vond ik nog de geringe invloed van de noise parameter. Zo lang deze niet zo groot is, dat grenzen gaan vervagen, heeft de exacte waarde ervan weinig invloed op de clustering. Het lijkt er dus op dat het toepassen van de normaal verdeelde ruis een toelaatbare techniek is, in dit geval.

Dit kleine experiment is vooral bedoelt om te laten zien dat en hoe de clustering toepasbaar is op data zoals die van het model te verwachten is. Voor een uitgebreidere analyse van de methode verwijs ik naar het originele artikel hierover (Carman & Merickel, 1990).

3.2 Convergent gedrag

Omdat het de bedoeling is dat het model sterke convergente eigenschappen heeft en binnen dat kader ook een aantal divergente eigenschappen moet kunnen verklaren, wordt in de eerste experimenten uitgegaan van een willekeurige initialisatie van het taalgebied. Er worden parameters aangepast om te laten zien hoe deze invloed hebben op de convergentie.



Figuur 3.1: Entropie voor experiment 1(1) ($\hat{\tau} = 0.3$, $\alpha = 3.0$)

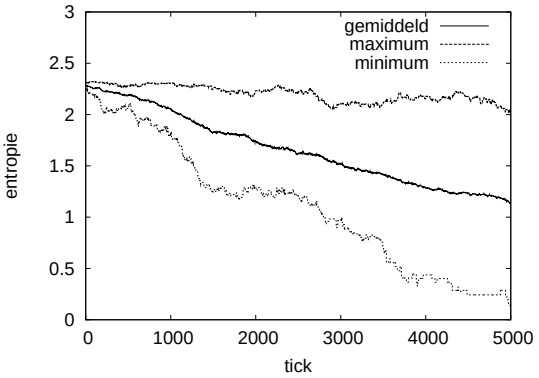
3.2.1 Experiment 1

In dit experiment staan de meeste model-parameters op hun standaardwaarde. De gedragingen mutatie, paniek en divergentie zijn uitgeschakeld.

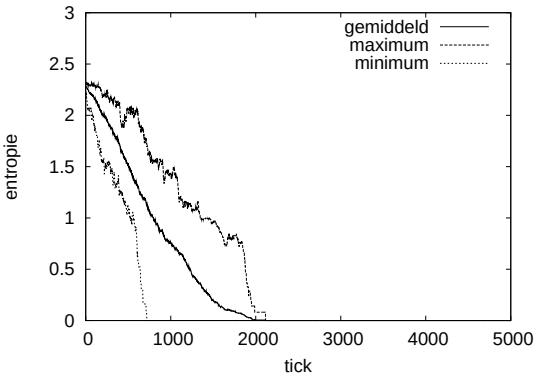
Het blijkt dat in dit geval, vanwege de grenswaarde instelling $\hat{\tau} = 0.3$, dat hoewel er een aantal grenzen wegvallen, het model convergeert naar een oplossing met veel zeer kleine (en behoorlijk gespreide) clusters. De minimale entropie loopt wat lichtelijk af, terwijl de maximale entropie vrijwel gelijk blijft. De gemiddelde entropie kent een zeer lichte daling (zie figuur 3.1). In het algemeen kan gezegd worden dat de entropie slechts op een klein aantal features nadert naar de minimale entropiewaarde.

Daarom is het interessant om te zien, of het model convergeert wanneer $\hat{\tau} = 1.0$. Wat opvalt is, dat er anders dan in het model van Axelrod, niet snel al te zien is dat er duidelijke clusters ontstaan waar het model lokaal al vrijwel geconvergeerd is. Het clusteringalgoritme is in het beginstadium dan ook niet erg nuttig om uitspraken te doen over de situatie. De gemiddelde entropie neemt zeer langzaam, maar gestaag, af (zie figuur 3.2). Gedurende de run vervagen de grenzen tussen de agents langzaam aan. Na zo'n 15×10^3 iteraties van het model, zijn er nauwelijks meer grenzen van betekenis en is de entropie voor de meeste features 0, of bijna 0 (het figuur bevat slechts 5×10^3 iteraties, zodat de schaal tussen de figuren gelijk is en de resolutie hoog genoeg).

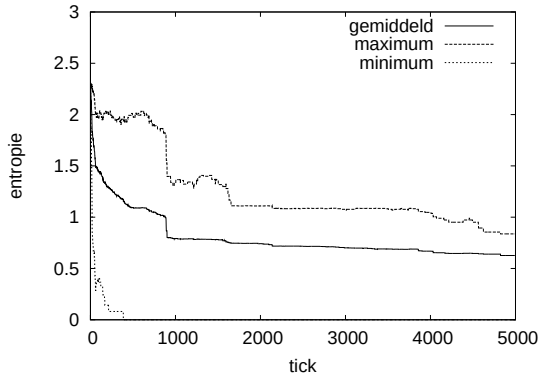
Door de stijlheid α van de convergentiekanscurve in te stellen op 0, kan dit proces aanzienlijk versneld worden. Na 1.0×10^3 is ongeveer eenzelfde niveau van convergentie bereikt als na 14×10^3 tijdstappen wanneer de curve een stijlheid $\alpha = 3.0$ heeft (zie figuur 3.3).



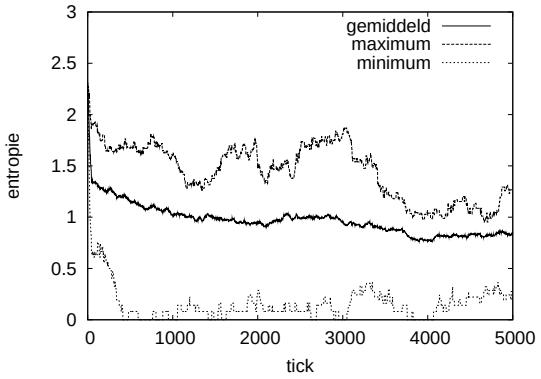
Figuur 3.2: Entropie voor experiment 1(2) ($\hat{\tau} = 1.0$, $\alpha = 3.0$)



Figuur 3.3: Entropie voor experiment 1(3) ($\hat{\tau} = 1.0$, $\alpha = 0.0$)



Figuur 3.4: Entropie voor experiment 2



Figuur 3.5: Entropie voor experiment 3

3.2.2 Experiment 2

In dit experiment zijn weer de standaardwaarden voor de parameters gebruikt. Dit maal staan de gedragingen mutatie en divergeren uit. Convergeren en paniek staan dus aan.

Het model gedraagt zich nu een stuk interessanter. In de eerste enkele tientallen tijdstappen gaat het model heel snel naar een situatie waar er een aantal gebieden zijn waarbinnen agents (vrijwel) dezelfde taal spreken. Er emergeren dus snel groepen agents die met elkaar kunnen praten uit de chaotische beginsituatie. Na dit voortvarende begin verandert er nauwelijks nog wat, tenzij twee gebieden zich door samenloop van omstandigheden kunnen verenigen. In dat geval is er een plotselinge, vrij scherpe val in de entropiewaarden waarneembaar (zie figuur 3.4). Merk hierbij op dat de eindsituatie (en daarmee het verloop van de entropie) hierbij, meer dan in het vorige experiment het geval was, er behoorlijk verschillend uit kan zien voor verschillende runs.

Het model is bij deze instellingen bijzonder veel sneller geconvergeerd dan in experiment 1. Echter, over het algemeen is er geen convergentie naar een situatie met maar 1 gebied.

Als de conformance parameter λ niet op de default ingesteld wordt, maar $\lambda = 0.8$ genomen wordt, is het gedrag heel anders: hoewel het begin nog steeds (anders dan in het model zonder paniek) gekenmerkt wordt door een scherpe daling in de entropie, vallen er niet snel grenzen weg.

Voor beide instelling geldt: na de beginfase is er nauwelijks meer sprake van geïsoleerde agents en neemt de convergentiegedraging het over. Wat nu opvalt, is dat er een veel homogenere situatie ontstaat dan met alleen de convergente gedraging en de default parameters, de verdeling is geografisch gezien veel samenhangender.

3.2.3 Experiment 3

Nu wordt ook de mutatie bij het gedrag betrokken, alleen het divergente gedrag wordt nog weggelaten. Wederom is begonnen met default parameterwaarden. Nu ontstaat er een situatie die vergelijkbaar is met de eerste situatie beschreven in het tweede experiment, maar met een wat dynamischer karakter. Binnen de gebieden ontstaan voortdurend vervagende of verscherpende grenzen.

Af en toe wordt een grens doorbroken, waardoor een gebied meestal alleen van vorm verandert. Er verdwijnen niet snel gebieden. Voor een plot van de entropie in een typische run, zie figuur 3.5.

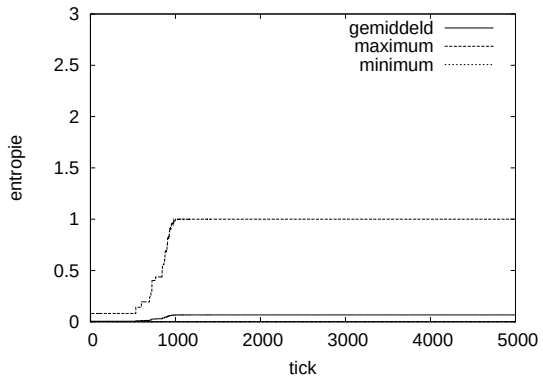
Wanneer de mutatiekans op het tienvoudige van de normale waarde gezet wordt, verdwijnen er in eerste instantie al meer grenzen, en verdwijnt er af en toe bij toeval een gebied wat daarvoor al vrij lang een stabiel bestaan leed. Zolang er geen scherpe grenzen doorbroken woden, loopt de entropie snel op. Wordt er eenmaal wel een grens doorbroken, dan wordt dit gekenmerkt door een plotselinge val van de entropie, vergelijkbaar met die in het begin.

3.3 Novelty drive

In deze experimenten wordt de 'novelty drive', het divergente gedrag in isolatie bekeken. Uitgangspunt is steeds een homogene situatie, waarbij voor één agent één feature een random waarde krijgt.

3.3.1 Experiment 4

In dit geval is gekozen voor de eerste methode bij de implementatie van het divergente gedrag. Er ge-



Figuur 3.6: Entropie voor experiment 5

beurt in dit geval, blijkt, erg weinig. Over een aantal verschillende runs was er geen zichtbare divergentie. Het is zelfs zo, dat de agent die de mutatie heeft, een relatief grote kans heeft om deze ongedaan te maken doordat juist die feature een hoge appealwaarde krijgt. Meestal convergeert het model dus (in ongeveer 100 tijdstappen) weer naar de homogene beginsituatie.

3.3.2 Experiment 5

Nu is hetzelfde experiment uitgevoerd, maar nu met de tweede methode. Ook nu gebeurt er in eerste instantie weinig, maar is er wel sprake van divergentie. Deze komt echter vrij traag op gang. Zodra een aantal agents de divergente eigenschap over hebben genomen, komt het echter op gang en nemen meer agents de eigenschap over. De entropiecurve lijkt op een sigmoïde: eerst een langzame toename, dan een veel snellere om vervolgens te convergeren naar een grenswaarde (zie figuur 3.6).

Als die grens bereikt is, bestaat de wereld ongeveer half-half uit agents met en zonder de mutatie. Deze wisselen elkaar af in langgerekte, aaneengesloten gebieden.

De divergentie heeft echter in de beginfase nog zeer duidelijk opstartproblemen. De vraag is dan ook, of de methode van partnerselectie wel tot een divergentie leidt die snel genoeg is. We spreken over enkele tientallen tot soms honderden tijdstappen voordat de divergentie op gang komt. Voordat hier verder op ingegaan wordt, wordt echter eerst nog nader naar het gedrag gekeken.

3.3.3 Experiment 6

In dit experiment is in het begin niet één, maar twee mutaties geïntroduceerd. De mutaties hebben betrekking op verschillende features. Hoewel de onset van divergentie behoorlijk kan verschillen, zullen meestal eerst per mutatie kleine gebieden ontstaan waar een dialect gesproken wordt. Deze breiden zich enigszins uit, tot ze elkaar raken. Nu gebeurt er in eerste instantie niet veel, maar op den duur ontstaan ook gebiedjes die beide mutaties bevatten.

Het kan natuurlijk ook gebeuren dat een agent met een mutatie eerst 'opgeslokt' wordt door een gebied met de andere mutatie en dat daarna pas de divergentie voor die mutatie op gang komt.

Nu bewijst de clustering-visualisatie zijn nut, want uit de lokale informatie valt weinig meer op ta maken: overal kleine en iets grotere verschillen, maar groepen vallen er moeilijk uit te halen. Uiteindelijk ontstaat er een grote ongeorganiseerde brei bestaande uit de vier mogelijke clusters: geen mutatie, 1 van beide mutaties (x 2), of allebei.

3.3.4 Experiment 7

Er wordt nu gekeken naar een alternatieve manier van partnerselectie: de agent kiest nu alleen uit de verzameling agents (in zijn eigen cluster), die ook daadwerkelijk minstens één feature verschillend hebben.

Het blijkt nu inderdaad dat de divergentie een stuk sneller gaat. Er lijkt echter kwalitatief zeer weinig verschil te bestaan, het eindresultaat ziet er vrijwel hetzelfde uit, maar wordt nu in enkele tientallen stappen bereikt.

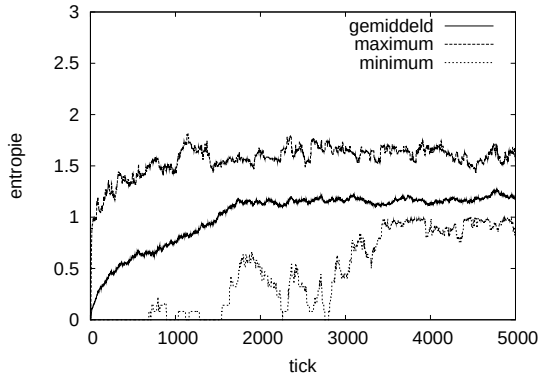
3.4 Taalverschijnselen

In de komende experimenten wordt gewerkt met het volledige model, waarbij voor het divergente gedrag methode 2 gekozen is, partnerselectie gebeurt alleen uit agents die daadwerkelijk verschillen van de agent zelf.

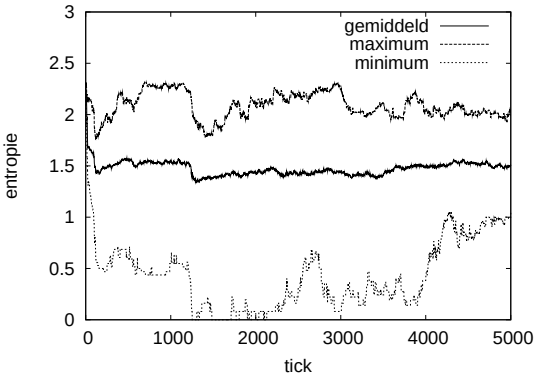
3.4.1 Experiment 8

Voor dit experiment zijn alle parameters ingesteld op hun standaardwaarde. Het model begint in een homogene situatie.

In het begin is dezelfde snelle toename in maximale entropie te zien als bij experiment 7, om de eenvoudige reden dat ook nu een mutatie snel verspreid dankzij het divergente gedrag. Die verspreiding krijgt echter niet de kans om duidelijke gebieden te vormen



Figuur 3.7: Entropie voor experiment 8



Figuur 3.8: Entropie voor experiment 9

zoals in experiment 7, maar wordt al snel verstoord door nieuwe mutaties.

Er ontstaat een gedrag dat lijkt op een extreme vorm van het gedrag in experiment 6, een wereld ontstaat waarin bijna iedereen een enigszins andere taal spreekt. Anders dan in experiment 6 krijgt het clusteringalgoritme weinig greep op de data en zijn de grenzen tussen agents bijna overal even scherp.

De entropie stijgt tot een grenswaarde die significant lager ligt dan de entropiewaarden van een willekeurige beginsituatie (zie figuur 3.7). Er is dus sprake van een gematigde taaldiversiteit.

Hier is dus nog geen sprake van een gradiënt, maar lijkt het meer op een gemeenschap waarin verschillende agents er enigszins een eigen stijl op nahouden. Het convergente gedrag houdt de verschillen blijkbaar goed genoeg in toom, dat er geen nieuwe grenzen ontstaan.

3.4.2 Experiment 9

Ook voor dit experiment zijn de standaardwaarden voor de parameters genomen, dit maal met willekeurig geïnitialiseerd taalgebied.

De convergente gedraging zorgt in dit geval voor een in eerste instantie vrij snelle daling in de entropie (zie figuur 3.8). Op den duur ontstaan een aantal min of meer stabiele gebieden, die lang kunnen blijven bestaan. Na de snelle daling van de entropie in het begin, blijft deze daarna vrij stabiel. Dankzij de aanwezigheid van random mutaties, blijft de kans natuurlijk altijd bestaan dat twee gebieden in elkaar over gaan, maar als de gebieden groot zijn, is de kans hierop erg klein.

Binnen de gebieden kan de variatie nog wel zo groot worden, dat het clusteringalgoritme binnen de duidelijk afgebakende clusters toch onderscheid maakt tussen een aantal 'subculturen'.

Dit experiment is verschillende keren uitgevoerd om te bepalen hoeveel taalgebieden er na 5000 ticks nog bestaan. Indien de grenzen niet direct uitsluitsel gaven over taalgebieden, werd het verder uitgevoerd met alleen het convergente gedrag ingeschakeld. Het aantal gebieden dat overbleef na volledige convergentie is het aantal taalgebieden naar 5000 runs. Het blijkt dat na tien dergelijke runs precies de helft eindigt met één taalgebied en de andere helft met twee.

Daarbij moet opgemerkt worden dat de situatie na 2000 ticks ook al vrij stabiel is, maar de kans vrij reëel is dat er na die tijd nog een (kleiner) gebied verdwijnt. Na 2000 ticks zouden de getallen dus iets hoger uitvallen. De kans is aanwezig dat voor 10000 ticks het aantal nog iets lager zou zijn.

4 Conclusie

In de inleiding wordt het begrip 'dynamische taaldiversiteit' geoperationaliseerd (zie sectie 1.1). Een nadeel van deze operationalisatie is dat deze kwalitatief van aard is, en daarmee moeilijk te meten. Het zou erg nuttig zijn om voor deze kwalitatieve criteria een standaard kwantitatieve meetmethode op te stellen.

4.1 Clustering

Het gebruik van een clusteringalgoritme voor de analyse van de data is een poging in die richting. Het gebruikte algoritme (zie sectie 2.2.3) is echter lang niet in alle situaties ideaal (zie sectie 3.1). De clustering is echter wel een stap op weg naar de analyse van situaties waarin lokale afstanden (zie sectie 2.2.2) niet voldoende informatie bieden.

Het gebruikte clusteringalgoritme maakt het al mogelijk om in meer situaties te visualiseren wat er gebeurt. De interpretatie van de visualisatie is echter niet helemaal triviaal, zoals bij de evaluatie van de clustering al duidelijk wordt (zie sectie 3.1). Door rekening te houden met de lokale afstandsinformatie, kan de visualisatie nog wel op de juiste manier gebruikt worden, maar het algoritme is niet robuust genoeg om een kwantitatieve operationalisatie van de benodigde verschijnselen mogelijk te maken.

4.2 Verschijnselen

De experimenten 1, 2, 3 en 9 laten zien dat het model in weze een convergent model is, dat wil zeggen - er is een sterke drijfveer naar een wereld waarin aaneengesloten gebieden bestaan waarin vrijwel dezelfde taal gesproken wordt.

In een model waarin mutaties opgenomen zijn, geldt voor alle grenzen dat de kans bestaat dat ze aangetast worden door een combinatie van mutaties en convergent of divergent gedrag. Afhankelijk van de grootte van het gebied, kan dit op termijn betekenen dat het opgeslokt wordt door een (groter) gebied, of dat er een voortdurend verplaatsende grens ontstaat.

Experimenten 4, 5, 6 en 7 maken duidelijk dat de novelty drive er inderdaad voor kan zorgen dat een enkele mutatie snel door andere agents in het model overgenomen wordt. Dit is dus een plausibele oplossing voor het 'threshold problem' (Nettle, 1999), waarbij geen noodzaak is voor het bestaan van hyperinvloedrijke agents.

In experiment 8 zorgen de convergente eigenschappen in combinatie met de novelty drive er binnen een homogeen taalgebied voor dat er individuele verschillen in taalgebruik ontstaan. Er ontstaan af en toe 'subculturen', die geografisch maar beperkte samenhang vertonen, zich verspreiden en weer opgaan in de massa. De taal als geheel is dus voortdurend aan verandering onderhevig.

De inleiding specificeert een aantal verschijnselen die het model moet laten zien (zie sectie 1.1). Uit experiment 8 blijkt, dat het eerste fenomeen inderdaad waargenomen wordt. Hoewel er geen nieuwe echte taalgrenzen ontstaan, is er wel voortdurend sprake van nieuwe 'dialecten'.

Ook aan de tweede voorwaarde is voldaan, zoals uit experiment 9 blijkt, zijn echte taalgebieden (met harde grenzen) alleen mogelijk als deze vrij groot zijn. Kleine gebieden zijn gedoemd op te gaan in grotere. Agents zonderen zich dus niet af in kleine groepen en er ontstaan geen 'kluizenaars'.

In hetzelfde experiment is ook te zien dat er over taalgrenzen voortdurend getwist wordt - deze blijven zich verplaatsen en aanpassen. Wordt een grens voldoende aangetast dan kan deze in zijn geheel verdwijnen. Fenomeen 3 is dus ook in het model aanwezig - grenzen veranderen voortdurend.

Fenomeen vier, het ontstaan van een 'dialectgradiënt', valt enigszins te herkennen in experiment 8, maar het is lastig hier daadwerkelijk conclusies over te trekken omdat zowel de lokale afstanden als het clusteringalgoritme hiervoor niet helemaal toereikend zijn.

Waarschijnlijk is er in dat experiment meer sprake van kleine individuele verschillen dan dat er een echte gradiënt zichtbaar is. Mogelijk heeft dit ook te maken met de keuze voor een toroïdaal grid, waarin een gradiënt niet gewoon van een uiteinde van het gebied naar een andere kan lopen. Het zal tussen twee punten moeten bestaan, waarbij het alle richtingen op bestaat - het is moeilijk hier een precieze voorstelling van te maken. Daarmee is het ook moeilijk om met zekerheid een gradiënt aan te wijzen. De analysemethoden schieten hier dus wat tekort.

4.3 Taaldiversiteit

Het model, met standaardinstellingen, voldoet dus aardig aan de in de inleiding gestelde operationalisaties van taaldiversiteit. De vraag is echter in hoeverre de gebruikte mechanismen plausibel zijn en daarnaast in hoeverre de gekozen operationalisaties realistisch zijn.

Ten eerste is het gebruik van een grotere neighborhood dan in het model van Axelrod (1997) mogelijk realistischer, er zijn niet veel mensen die slechts 4 anderen regelmatig genoeg spreken om door hen beïnvloed te worden in het taalgebruik. Dit model gebruikt een groep van 24 anderen. Een model waarin interacties met alle andere agents (op gelijkwaardige basis) opgenomen zijn, is ook weer niet realistisch.

Het mechanisme 'mutatie' is te verklaren in termen van productiefouten van agents (als een agent consequent een 'fout' gaat maken), die door andere agents dan eventueel overgenomen kan worden. Gebeurt dit niet, dan kan het ook zijn dat de agent zijn fout hersteld.

Het 'paniek' mechanisme is gebaseerd op de aanname dat geen enkele agent lang kan overleven zonder te communiceren met zijn omgeving. Kan de agent dus met onvoldoende andere agents communiceren dan zal deze agent in paniek raken en zijn best doen een andere taal te leren.

De convergente gedraging komt overeen met het belang voor de agent om door zijn omgeving begrepen te worden en om zijn omgeving te begrijpen. Daarom zal een agent in grote lijnen conformeren aan de in zijn omgeving gesproken taal.

Daarnaast is er echter de 'novelty drive', die tegen het principe dat agents met elkaar willen kunnen communiceren in lijkt te gaan. Dit mechanisme correspondeert echter met een ander principe, namelijk de behoefte aan variatie in de eigen omgeving en de drang van de agent om zich te onderscheiden van zijn omgeving (maar natuurlijk niet te veel).

Het is dus wel degelijk mogelijk om de gepostuleerde mechanismen te projecteren op de werkelijkheid. De vraag of deze mechanismen ook op een vergelijkbare manier bestaan in de werkelijkheid is een heel andere, die hier ook niet beantwoord zal worden.

5 Discussie

5.1 Het nut van clustering

Het gebruik van lokale afstanden om gebieden te onderscheiden (zoals ook Axelrod (1997) doet) kan een belangrijke reden zijn om een von Neumann neighborhood van vier agents te gebruiken, een grotere neighborhood zou de analyse te moeilijk maken. Dit is een duidelijke beperking om de mogelijke modellen die naar mijn mening niet altijd wenselijk is.

De gebruikte clusteringalgoritme is een poging om dit op te lossen, die deels geslaagd is. Het is echter niet geschikt om te gebruiken in een kwantitatieve operationalisatie van dynamische taaldiversiteit.

Daarvoor zou in iedere geval een algoritme nodig zijn dat werkt met vaste thresholds, die mogelijk afhangen van de modelparameters. Alleen dan kan met behulp van de clustering uitspraken gedaan worden die tussen verschillende runs van het model en tussen verschillende experimenten vergelijkbaar zijn. Waarschijnlijk is het nuttig om dit voor verschillende thresholdwaarden te doen, zodat je onderscheid kan maken tussen 'cultuur' en 'subcultuur' niveau.

Het is opvallend hoe lastig het is om een goede vergelijking van clusteringalgoritmen te vinden. Overzichtsartikelen zijn vaak erg informeel en laten weinig resultaten op verschillende datasets zien. Artikelen over een (verbetering op) een algoritme vergelijken vaak met maar één ander algoritme. Het zou dus handig zijn om een verzameling datasets te hebben, met een verzameling bijbehorden technieken voor het evalueren van de prestaties van een clusteringalgoritme.

Het lijkt er echter op dat er wat dit betreft geen sprake is van standaardisatie en dat elke auteur zijn eigen evaluatie bedenkt en uitvoert. Dat is ook wat in deze these gedaan is voor de evaluatie van het ISODATA algoritme, iets wat naar mijn mening de waarde, relevantie en betrouwbaarheid van een dergelijke evaluatie niet ten goede komt.

Kortom, het zou zeker nuttig zijn om meer onderzoek te doen naar de analyse van dit soort modellen met behulp van clusteringalgoritmen. Het gebrek aan een goede (standaard-) methode hiervoor is een behoorlijke beperking op de mogelijke modellen en maakt het vergelijken van resultaten lastig.

5.2 Verdere experimenten

De experimenten zijn alleen uitgevoerd met een 10×10 wereld, iedere agent gaat dus om met $\frac{1}{4}$ van de wereld. Dat is niet een erg realistische aanname en het zou dus zeker nuttig zijn om dezelfde experimenten te herhalen met een (veel) grotere wereld. Echter, dat is ook computationeel veel duurder en zou waarschijnlijk erg veel tijd kosten.

Eenzelfde argument geldt overigens voor veel andere parameters, de parameterruimte is erg groot en over het algemeen zijn de experimenten slechts uitgevoerd voor de standaardwaarden. Door hiermee te spelen zouden nog andere taalfenomenen wellicht nagebootst kunnen worden.

Met de huidige instellingen ontstaan in een homogene situatie niet werkelijk nieuwe talen. Mogelijk zijn er parameterwaarden waarvoor dat wel het geval is. De vraag is echter wel, of dat realistisch zou zijn zonder dat er een externe drijfveer (zoals geografische afzondering) voor is.

Wat betreft de gekozen operationalisaties, die lijken zeker nog niet volledig te zijn. Hoewel het model al meer verschijnselen kan verklaren dan veel eerdere modellen, lijkt het waarschijnlijk nog niet erg sterk op 'echte' taalverandering. Daarnaast is het vervelend dat er geen kwantitatieve operationalisaties beschikbaar zijn.

Referenties

Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *The Journal of Conflict Resolution*, 41(2):203–226.

Ball, G. H. en Hall, D. J. (1965). Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, Springfield, USA.

Barr, D. J. (2004). Establishing conventional communication systems: Is common knowledge necessary? *Cognitive Science*, 28(6):937–962.

Buzing, P. C., Eiben, A., en Schut, M. C. (2005). Emerging communication and cooperation in evolving agent societies. *Journal of Artificial Societies and Social Simulation*, 8(1).

Carman, C. en Merickel, M. (1990). Supervising isodata with an information theoretic stopping rule. *Pattern Recognition*, 23(1/2):185–197.

Jain, A., Murty, M., en Flynn, P. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(327):86–101.

Nettle, D. (1999). Using social impact theory to simulate language change. *Lingua*, 108(2-3):95–117.

North, M., Collier, N., en Vos, J. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1–25.

Pearson, J. E. (1993). Complex patterns in a simple system.

Railsback, S. F., Lytinen, S. L., en Jackson, S. K. (2006). Agent-based simulation platforms: Review and development recommendations. Manuscript re-submitted to Simulation.

Schäfer, J. en Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4.

Wang, W. S.-Y. en Minett, J. W. (2005). The invasion of language: emergence, change and death. *Trends in Ecology and Evolution*, 20(5):263–269.

Weisstein, E. W. (2003a). Moore neighborhood. *MathWorld–A Wolfram Web Resource*, page http://mathworld.wolfram.com/MooreNeighborhood.html.

Weisstein, E. W. (2003b). von neumann neighborhood. *MathWorld–A Wolfram Web Resource*, page http://mathworld.wolfram.com/vonNeumannNeighborhood.html.

Weisstein, E. W. (2007). Entropy. *MathWorld–A Wolfram Web Resource*, page http://mathworld.wolfram.com/Entropy.html.