

INTERACTIONS OF FORMING LANGUAGES

Project Report — Modeling and Simulating Social Systems with MATLAB

written by

Heim W. Steve and Wochner David

Zurich, May 23 2010



Contents

1	Introduction & Motivation	2
2	Description of the Model	4
3	Implementation	6
4	Results and Discussion	17
4.1	Case I	17
4.2	Case II	18
4.2.1	1 vs 1	18
4.2.2	2vs2	19
4.2.3	3vs3 and 5vs5	19
4.3	Case III	20
4.3.1	5vs1	20
4.3.2	5vs2	21
4.3.3	3vs2	22
4.4	Case IV	22
4.4.1	3v2, Blue 5 Red 15	23
4.4.2	5v1, blue 1 red 19	23
4.4.3	3v2, blue 12 red 8	23
5	Conclusion and Outlook	25
5.1	Conclusion	25
5.2	Outlook	25
	Bibliography	26

1 Introduction & Motivation

In nature several methods of communicating can be found: ants and bees both live in relatively complex societies with a distinct hierarchy, however communication bases itself primarily on instinctive reactions to emitted biochemicals, and even each class is defined by very different genetic and therefore anatomic differences. On the other hand amongst higher mammals communication is much less a built-in behavior and prevalently takes the form of body language, such as bearing ones teeth, however the social structures of higher mammals is a much simpler, like the alpha male in a pack of wolves.

For a structured and complex society composed of independent individuals such as created by humans, a common way of communication is quintessential. The prime attribute for such a language is that amongst a certain tribe of individuals, a set of syntax and rules are laid down by which one person can express himself so that another person can successfully understand his meaning. But how is a language born? How do individuals, with only rudimentary and unspecific means of communicating, agree to which words should mean what? Somehow, somewhere along the way, humans started to speak with one another and develop and fix these rules. Several studies ([4]-[6]) on this phenomena have been conducted already, both by studying the syntax of existing and extinct languages (from the top down) as well as by simulating the formation process (from the bottom up). In this report, the literature that attempt to simulate language formation by means of a mathematical model is of primary interest. One of the most prominent such models is the Steel[1] language game, which simulates single conversations between two individuals, a speaker and a listener. The speaker attempts to communicate a certain message for a certain object to the listener, like a person holding out a bone and saying "bone!". If the listener agrees on the message being used for this object the conversation has taken place and the two individuals understand each other. Otherwise the conversation results as a failure, like a person shaking his head and saying "no, Knochen!".

A student of ETH Zurich, Martin Heynen[2], recreated this model using MATLAB to model the formation of single languages. In this report, the interaction between different relatively young languages is examined, focusing particular on the influence of one language on the development of a second language. To this end, Heynen's model has been enhanced to handle multiple languages as well as give languages a sense of duration (i.e. languages that have been around longer tend to be more robust and resistant to outside influence).

This is of particular interest to understand languages of populations that have been subjugated by another population with a different language, or otherwise undergone heavy influence from another language. Example of this are Germanic and Gallic languages during the Roman expansion, as well as the languages of most Eastern Asian populations neighboring China. English is also a great example, for despite coming from a Germanic root has a predominantly latin-derived

vocabulary, mostly due to English relations with France.

2 Description of the Model

The Heynen model describes a population of agents with the following interaction characteristics (for a more detailed description see Heynen's report, attached):

-Language composed of N objects and N messages: These messages are correlated to their corresponding object through a matrix.

$$[\text{matrix example}] = \begin{array}{c} \text{objects} \\ \left[\begin{array}{ccc} 5 & 1 & 1 \\ 20 & 25 & 20 \\ 1 & 1 & 1 \end{array} \right] \end{array} \begin{array}{c} \text{messages} \end{array}$$

The minimum value of any element of this matrix is 1, with no value cap. The higher the relative value of a certain element, the stronger the link between that message and object, so in the example, the connection between message 1 and object 1 is strong, however the connection between message 2 and object is not so strong, and object 3 obviously doesn't have any specific corresponding message. It is important to note that message-object correlations aren't mutually exclusive per se, so a message could potentially have 2 meanings, and 2 messages could potentially mean the same thing. However this never happens because of how the update function is implemented (see below). The reason the relative value is important is the the presence of a small random factor when choosing the corresponding message, meaning the message with the highest value is simply the most likely to be chosen.

- Population of X agents

For each simulation a fixed population was used in order to keep results uniform and coherent. Related to the model, this means there are X Language matrices, one for each agent, or rather a 3-dimensional matrix of objects, messages and agent.

- iteration of matchrounds

In each matchround, each agent randomly meets another agent and has a conversation. The speaker randomly picks an object to communicate, then tells the listener which object he means and which message he thinks corresponds to it. The listener then checks which message he thinks should correspond to that object and tells this to the speaker. Then the language matrices of both agents are updated depending on whether the conversation was successful (both agents picked the same message for the object) or not. For the exact update mechanisms please see Heynens report.

To extend the Heynen model to model interactions between more than 1 language it was of course first of all necessary to enlarge the language matrix to handle more than 1 message per object, resulting in something like this:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

In this example it is quite obvious that more than one language is possible, and in fact 2 completely different languages are possible, as well several combinations or "dialects". In order to see the influence of one language on another, it is of course necessary to give the matrix starting values at the beginning of the simulation:

$$[\text{blue agent}] = \begin{bmatrix} 5 & 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 5 & 1 & 1 & 1 \end{bmatrix}$$

$$[\text{red agent}] = \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

In this example, both blue and red agents already have specific and different messages corresponding to each object, thereby giving them 2 different starting languages. However, the starting value for the blue language (5) is much higher than that of the red language, thereby indicating that the blue language is more developed integrated into that population, whereas the red language is 'younger' so as to speak.

3 Implementation

Below the the entire model can be found in MATLABcode in two files, 'LanguageInteractor.m' and 'newUpdate.m'. Although similar to the Heynen's code, several important changes have been made (for exact changes, please compare with Heynen's code). Most importantly:

- Added functionality to handle multiple languages; the biggest change, this is involved the entire code, as well as modifying the update function slightly.
- The entire code has been embedded into a function to make calling data easier. Please use "help" (F1) to see how to use the function.
- variable names and code in general has been rendered somewhat more readable
- Added functionality to handle multiple languages
- Drastic updates I, II and III have been commented out as they do not make sense with the model this report aims at. They have however been left in the code to allow the user to test single-language dynamics with the same code as well (maintains all previous functionality - backwards-compatible).
- Added 2 bar plotters within the code to facilitate data generation.

```
1 %%
2 % Model on language interactions by Steve W Heim and David Wochner
3 % developed for the class "Modelling social systems with MATLAB"
4 % based on model and code by Martin Heynen.
5 %
6 % Please make sure to include the file "update.m"
7 %
8 % function LanguageInteractor(numSimulations, numRounds,
9 % StartValBlue, StartValRed)
10 %
11 % INPUTS:
12 % numSimulations: number of times you want to run the simulation. [int]
13 % numRounds: matchrounds that should be iterated (after -100 stable) [int]
14 % blueAgents: number of blue agents
15 % redAgents: number of red agents
16 % StartValBlue: starting value for Blue language (generally between 1-5)
17 % StartValRed: starting value for Red language [double]
18 %
19
```



```
20
21 %%
22 function LanguageInteractor(numSimulations, numRounds,blueAgents,...
23     redAgents, startValBlue, startValRed)
24
25
26 for simulation=1:numSimulations
27
28 % Only for running outside function (direct copy/paste)
29 % clear
30 % clc
31 % numRounds=1000;
32 % START VALUE OF PROBABILITY MATRIXES A AND B
33 % StartValBlue = 1;
34 % StartValRed = 1;
35 % blueAgents = 10;
36 % redAgents = 10;
37
38 % INITIAL SETUP
39 % POPULATING AGENTS
40
41 agents = blueAgents + redAgents;
42
43 % SETTING UP LANGUAGES
44 numberObjects=3;
45 numberMessages=6;
46
47 % Only necessary when using drastic update 1
48 % emergencyBrake = agents*(agents-1);
49 % partOfAgents=3;
50
51 conversationhistory=0;
52 A=ones (numberObjects,numberMessages,agents);
53 B=ones (numberMessages,numberObjects,agents);
54
55 for alpha = 1: blueAgents
56     for i=1:numberObjects
57         A(i,i,alpha) = startValBlue;
58         B(i,i,alpha) = startValBlue;
59     end
60 end
61
62 for beta = (blueAgents+1):agents
63     for i=numberObjects+1:numberMessages;
64         A(i-numberObjects,i,beta) = startValRed;
65         B(i,i-numberObjects,beta) = startValRed;
66     end
67 end
68
69
70
```

```

71 firsttime100percent = NaN;
72 drastic=0;
73 %successfulConversation = 0;
74
75 %ROUND WHERE EVERY AGENT MEETS EVERY AGENT
76 %AT THE END OF EVERY ROUND MATRIX A AND B GET UPDATED ACORDING
77 %TO THE MEMORY. THE MEMORY STARTS FROM ZERO EVERY ROUND
78 for matchround=1:numRounds
79
80     conversation=0;
81     %counter for conversation success
82     MA=zeros(numberObjects,numberMessages,agents);
83     %Memory from the sender gets set to zero
84     MB=zeros(numberMessages,numberObjects,agents);
85     %Memory from the listener gets set to zero
86     %for the drastic update III
87     % convcheck=zeros(1,agents);
88     %AGENT MATCH
89     for agentone=1:agents
90         for agenttwo=1:agents %first agent meets all the others...
91             if agentone≠agenttwo%no monologues
92
93                 senderagent=agentone;
94                 listeneragent=agenttwo;
95
96
97                 %RANDOM PICKED OBJECT FROM 1 TO NUMBER OF OBJECTS
98                 object=round(0.5+(numberObjects+0.4999-0.5)*rand(1,1));
99                 %0.5-3.4999 gerundet! => uniform 1-#objects
100
101                 %SENDER: MESSAGE SENT FOR OBJECT, CHOSEN ACCORDING
102                 %PROBABILITY OF MATRIX A
103                 rowsumA=0;
104                 for q=1:numberMessages
105                     rowsumA=rowsumA+A(object,q,senderagent);
106                 end
107                 c=rand(1,1)*rowsumA;
108                 rowsum=0;
109                 for i=1:numberMessages
110                     rowsum=rowsum+A(object,i,senderagent);
111                     if c≤rowsum
112                         message=i;
113                         break
114                     end
115                 end
116
117                 %LISTENER UNDERSTANDS OBJECT FROM RECIEVED MESSAGE
118                 %ACCORDING PROBABILITY OF MATRIX B
119                 rowsumB=0;
120                 for q=1:numberObjects
121                     rowsumB=rowsumB+B(message,q,listeneragent);

```

```

122         end
123         d=rand(1,1)*rowsumB;
124         rowsum=0;
125         for i=1:numberObjects
126             rowsum=rowsum+B(message,i,listeneragent);
127             if d<=rowsum
128                 understandsobject=i;
129                 break
130             end
131         end
132     end
133
134     %MEMORY: AFTER EVERY CONVERSATION THE SENDER TELLS THE
135     %LISTENER WHAT OBJECT HE MEANT AND BOTH SAVE THAT IN
136     %THEIR MEMORY. AFTER A SUCCESSFUL CONVERSATION THE
137     %INCREMENT IS AS TWICE AS HIGH.
138     increasemem = 1;
139     if understandsobject==object
140         conversation=conversation+1;
141         %successful conversation counter
142         increasemem = 2;
143     end
144
145     end
146
147
148
149     MA(object,message,senderagent) = MA(object,...
150         message,senderagent)+increasemem;
151     MB(message,object,listeneragent)=MB(message,...
152         object,listeneragent)+increasemem;
153
154     %%
155     %DRASTIC UPDATES
156     %-----
157     % %DRASTIC UPDATE I
158     % %Note that this is not used in this model since we
159     % %actually want languages to become more developed and
160     % %deep rooted amongst a certain population over time.
161     % %This simulates "old" and "young" languages.
162     %
163     % %IF THE SENDER IS VERY SURE (PROB. FOR MESSAGE 100 UNITS
164     % %BIGGER THAN FOR ALL OTHER MESSAGES)
165     % %TO SEND THE MESSAGE THE LISTENER
166     % %WILL UNDERSTAND AND EVEN THOUGH THE CONVERSATION FAILS
167     % %THE PROBABILITY OF THE MESSAGE GETS REDUCED
168     % %DRASTICALLY TO THE SECOND HIGHEST PROBABILITY
169     %     if understandsobject<object
170     %         emergencyBrakehelp=0;
171     %         for col=1:numberObjects
172     %             if col<=message

```

```

173 % AA=A(object,message,senderagent)-A(object,col,senderagent);
174 % if AA > emergencyBrake && AA < (emergencyBrake+agents/partOfAgents)
175 % A(object,message,senderagent)=A(object,message,...
176 %             senderagent)-(emergencyBrake);
177 %end
178 %end
179 %end
180 %
181 %
182 %end
183 %
184 %
185 %     %DRASTIC UPDATE II
186 %     %
187 %     %CHECKS AFTER EVERY CONVERSATION IF THE PROBABILITY FOR
188 %     %THE MESSAGE CHOSEN IN THIS ROUND IS HIGHER THAN 50 %
189 %     %AND IF YES IT CHECKS IF THIS IS ALSO TRUE FOR AN OTHER
190 %     %VALUE IN THE COLUMN OF THAT MESSAGE AND IF YES IT SETS
191 %     %BOTH VALUES TO 1
192 %     %This is also not used in this model, since it would
193 %     %be triggered in a lot of cases when an agent is in
194 %     %the process of "unlearning" one language and learning
195 %     %a new one.
196 %
197 %         zsum = sum(A(:, :, senderagent)'); %row vector with row sum
198 %         if A(object,message,senderagent)/zsum(object) > 0.5
199 %             for row=1:numberObjects
200 %                 if row~=object
201 %                     if A(row,message,senderagent)/zsum(object) > 0.5
202 % A(row,message,senderagent)= min(A(row,message,...
203 %             senderagent),A(object,message,senderagent));
204 % A(object,message,senderagent)=min(A(row,message,...
205 %             senderagent),A(object,message,senderagent));
206 %                 break
207 %             end
208 %         end
209 %     end
210 % end
211 %
212 %
213 %
214 %     %DRASTIC UPDATE III.1
215 %     %
216 %     % CHECKS IF IN ONE MATCHROUND ALL THE CONVERSATION OF AN
217 %     % AGENT IN THE ROLE AS A THE SPEAKER FAILED AND IF YES
218 %     % THEN IT SETS BOTH OF HIS PROBABILITY MATRIXES TO THE
219 %     % STARTVALUE
220 %     % Again, this drastic update would interfere with this
221 %     % model, since it could undo all the "unlearning" and
222 %     % "learning" of an agent, therefore negating the
223 %     % influence of one language on the other.

```

```

224 %         if understandsobject≠object
225 %             convcheck(1,agentone)=convcheck(1,agentone)+1;
226 %
227 %         end
228 %-----
229
230
231         end
232     end
233
234     end %agentone
235
236     %UPDATE OF THE MATRIXES ACCORDING TO MEMORY
237     %see update function
238     for agent=1:agents
239         A = newUpdate(MA, A, numberObjects, numberMessages, agent);
240         B = newUpdate(MB, B, numberMessages, numberObjects, agent);
241     end
242     conversationhistory(matchround,1)=conversation;
243
244
245
246
247     if conversation == agents*(agents-1)
248         firsttime100percent = matchround;
249     break
250     end
251
252 %     %%DRASTIC UPDATE III.2
253 %
254 %     for y=1:agents
255 %         if convcheck(y)≥agents-1
256 %             drastic=drastic+1;
257 %             A(:, :, y)=startvalue;
258 %             B(:, :, y)=startvalue;
259 %         end
260 %     end
261 %
262 %AFAIL SHOWS HOW THE MATRIXES LOOK WHEN IT DID'T REACH 100% IN THE GIVEN
263 % NUMBER OF ROUNDS
264 %-----
265 % if matchround==numRounds
266 %     st=st+1;
267 %     Afail(:, :, :, st)=A;
268 % end
269 %-----
270
271 %% Time-evolution plotter
272 % This plotter gives a bar plot of the message values every 10 rounds,
273 % therefore giving a chronological view of the evolution of the languages.
274 % Be careful when using this plotter, it is advised to reduce the number of

```

```

275 % matchrounds to 100 (10 plots), also pay attention to the conflicts of the
276 % figure number if you're using other plots as well.
277
278 % if (mod(matchround,10) == 0)
279 %     for i = 1:agents
280 %         for j = 1:numberMessages
281 %             Val(i,j) = sum(A(:,j,i));
282 %         end
283 %     end
284 %     Val = Val./numRounds;
285 %
286 %     NVal = zeros(size(Val));
287 %     for i = 1:agents
288 %         for j = 1:numberMessages
289 %             if Val(i,j) > 0.2
290 %                 NVal(i,j) = 1;
291 %             end
292 %         end
293 %     end
294 %     figure(matchround/10);
295 %     bar3(Val)
296 %     axis([0 6 0 20 0 3]);
297 %     xlabel('Messages');
298 %     ylabel('Agents');
299 % end
300
301
302 end %matchround
303
304
305 %% Success Plotter
306 % This plotter plots the percentage of successful conversations per
307 % matchround. It is useful to see the "uniformity" of the developed
308 % languages and dialects.
309
310 figure(1);
311 hold on;
312 conversationhistorypercent=conversationhistory./(agents*(agents-1)).*100;
313 plot(1:matchround,conversationhistorypercent)
314 xlabel('matchround');
315 ylabel('successful conversations [%]');
316 ylim([1 100])
317
318 %%
319
320 %STATISTICS
321 %-----
322 % countesuc(z,1)=firsttime100percent;
323 % if firsttime100percent == NaN
324 %     A
325 %     B

```

```

326 %     conversation/(agents*(agents-1)).*100
327 % end
328 %
329 %
330 % end%for row 7
331 % countesuc(1,2)=nanmean(countesuc);
332 % countesuc(2,2)=nanmax(countesuc(:,1));
333 % countesuc(3,2)=nanmin(countesuc(:,1));
334 beep
335
336 %% Language plotter
337 % This plotter is also a bar plotter, but simply gives a unitary bar to
338 % messages with a high value (messages that get used), and is useful to see
339 % what message combinations (languages and dialects) actually resulted at
340 % the end of the simulation.
341
342 for i = 1:agents
343     for j = 1:numberMessages
344         Val(i,j) = sum(A(:,j,i));
345     end
346 end
347 Val = Val./numRounds;
348
349 NVal = zeros(size(Val));
350 for i = 1:agents
351     for j = 1:numberMessages
352         if Val(i,j) > 0.2
353             NVal(i,j) = 1;
354         end
355     end
356 end
357 figure(1 + simulation); %+1 to avoid conflict with the success plotter
358 bar3(NVal)
359 axis([0 6 0 20 0 3]);
360 xlabel('Messages');
361 ylabel('Agents');
362
363 end                %END OF SIMULATION
364
365 end                %END OF FUNCTION

```

```

1 function y = newUpdate(MA, A, rowNum, colNum, agent)
2
3 shift          = 1;                % update unit for prob. matrix
4 dist          = 1;                % update criteria
5
6 toUPA=zeros(rowNum,colNum);
7 onesezero=zeros(rowNum,colNum);
8
9

```

```

10 %MATRIX toUPA: IF IN THE MEMORY MATRIX THE BIGGEST VALUE IN A ROW IS
11 %(1+dist) BIGGER THAN THE OTHER VALUES IN THE ROW THE toUPA MATRIX IS SET
12 %TO 1 AT THIS POSITION
13
14 [CA, IA] = max(MA(:, :, agent)'); % IA coloum-position of the max for every row
15
16 for i=1:rowNumber %row i
17     for j=1:colNumber %coloum j
18         if j≠IA(i) %for all coloum-positions except where the max is
19             if MA(i, IA(i), agent) - MA(i, j, agent) ≥ dist + 1
20                 toUPA(i, IA(i))=1;
21             else
22                 toUPA(i, IA(i))=0;
23             break
24         end
25     end
26 end
27 end
28
29
30 %CHECKS IF THERE IS A MAX CONFLICT IN A COLOUM
31 colsumA = sum(toUPA); %row-vector with coloum-sums
32 conflict=0;
33 for i=1:colNumber
34     if colsumA(1, i)>1
35         conflict=1;
36     end
37 end
38
39
40
41
42 %IF THERE'S A CONFLICT: CHECKS IF THERE IS A REAL MAX IN THE COLOUM(1+dist)
43 %BIGGER THAN OTHER MEMORY ENTRIES)
44 %IF YES THEN UPDATE toUPA FROM 1 TO ZERO WHERE THE MAX IS NOT (onesetzero
45 %REMEMBERS WHERE THIS WAS DONE)
46 if conflict==1
47     % rowmax row position of the max for every coloum
48     [CA, rowmax] = max(MA(:, :, agent));
49     for col=1:colNumber
50         for row=1:rowNumber
51             if row≠rowmax(col)
52                 if MA(rowmax(col), col, agent) - MA(row, col, agent) ≥ dist + 1
53                     if toUPA(row, col)==1
54                         toUPA(row, col)=0;
55                         onesetzero(row, col)=1;
56                     end
57                 end
58             end
59
60 %IF THERE IS NO REAL MAX IN THE COLOUM ALL THE ONES OF toUPA GET 2's TO
61 %MAKE SURE THAT THERE WON'T BE AN UPDATE

```



```

61         else
62             for j=1:rowNumber
63                 if toUPA(j,col)==1
64                     toUPA(j,col)=2;
65                 end
66             end
67             break
68
69         end
70     end
71 end
72 end
73
74
75 %IF THERE'S NO MAX CONFLICT IN THE COLOUM
76 else
77 rowsum = sum(toUPA'); %rowvector with rowsums
78     for row=1:rowNumber
79         for col=1:colNumber
80             %only if there is a max in the row
81             if toUPA(row,col)==0 && rowsum(row)≠0
82                 A(row,col,agent)=A(row,col,agent)- shift;
83                 if A(row,col,agent)<1
84                     A(row,col,agent)=1;
85                 end
86             elseif toUPA(row,col)==1
87                 A(row,col,agent)=A(row,col,agent)+ shift;
88             end
89         end
90     end
91 end
92
93
94 toUPA;      %%%TEST
95 onestzero; %%%TEST
96
97
98 %UPDATE PROCESS IF MAX COLOUM CONFLICT
99 if conflict==1
100     rowsum = sum(toUPA');
101     for col=1:colNumber
102         for row=1:rowNumber
103             if rowsum(row) < 2 %only if there is a real max
104                 if toUPA(row,col)== 0 && rowsum(row) == 1
105                     %only if there is a zero and the row
106                     %has a max (not everythin zero)
107                     A(row,col,agent)=A(row,col,agent)- shift;
108                     if A(row,col,agent)<1
109                         A(row,col,agent)=1;
110                     end
111

```

```
112         elseif rowsum(row) == 0 && onesetzero(row,col)==1
113             %if zero and position where due to a
114             %real max in another row (but same coloum)
115             %the one was set to zero
116
117             A(row,col,agent)=A(row,col,agent)- shift;
118             if A(row,col,agent)<1
119                 A(row,col,agent)=1;
120             end
121
122         elseif toUPA(row,col)==1
123             A(row,col,agent)=A(row,col,agent)+ shift;
124         end
125     end
126 end
127 end
128 end
129 y = A;
130 end
```

4 Results and Discussion

In this report the interactions of two populations, Blue tribe and Red tribe, is studied. In order to maintain simplicity, all simulations have been run with 3 objects and 6 messages and a total population of 20 agents. Three different cases have been focused on:

1. In the first case, Blue tribe starts with an existing language, whereas Red tribe still has no language whatsoever.
2. In the second case, both tribes have pre-existing languages, both of the same "age" and therefore both with the same influence and susceptibility.
3. In the third case, Blue tribe will have an "older" and therefore more dominant language.
4. In the final case, the populations of both tribes will no longer be kept equal. This will be particularly interesting to study the case of minorities!

4.1 Case I

The first step in this experiment was to determine what range of starting values or language age was interesting to study. A higher value represents an old language which is therefore both more resistant to outside influence and more prone to influencing. Since it is precisely this influence that is the focus of this report, it is important to find out a threshold age after which influence of a language is negligible. To find this the influence of an old language (high starting value) on a population with no language (starting value all ones) was tested, and simulations with very high starting values were compared to simulations with low starting, the lower values gradually being increased till the difference in influential power was minimal.

As can be seen below, the the success rate between simulations with starting values of 5 and 100 is so close that it can concluded that the influence of all languages older than 5 exert the same amount of influence.

Taking a closer look at the actual results, the actual languages that formed can be seen:

Each color denotes 1 message type, ergo a set of 3 bars represents 1 language. Despite the blue population having a starting value twenty times greater in the simulation on the right, the red population still formed some of her own dialects. Although in this case it may seem that more dialects were formed in the left simulation (12 red bars compared to 9), over the large amount of simulations that were examined, the formation of dialects in both cases varied within the same range.

It is therefore trivial to conclude that the threshold of a language's potential influence is around 5.

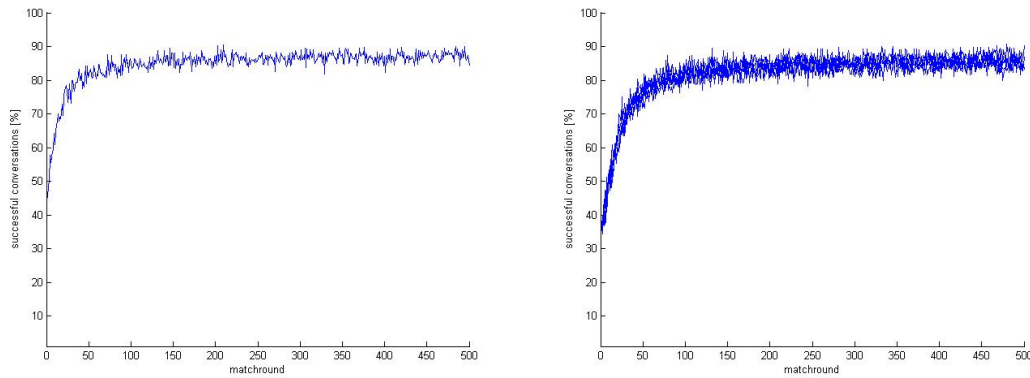


Figure 4.1 – The success graphs of the case with starting value of 5 and 100, left and right respectively. These measure the percentage of successful conversation per match round as time progresses.

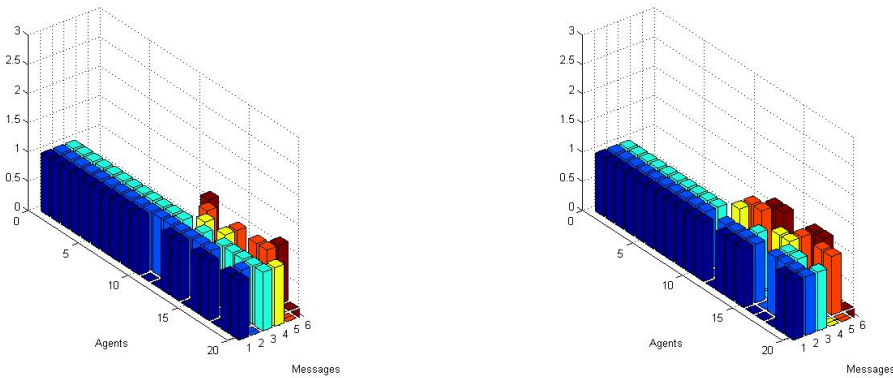


Figure 4.2 – Left graph is 100vs1, right graph is 5vs1.

Another conclusion that can be seen from the success graph is that the success rate stabilizes after roughly 100 rounds, meaning after 100 rounds languages are relatively fixed and don't change any more (resistance of 100%). As can be later seen, this tend to be true for all cases, regardless of starting values.

4.2 Case II

In this case, both languages have the same starting value as well as the same population. The results show how this bears on the final languages of the two tribes.

4.2.1 1 vs 1

In this run neither tribe has a starting language, meaning, from a linguistic point of view, there aren't two distinct tribes but rather one large tribe. This is the situation which Heynen's model simulates, with one big distinction: In Heynen's model only one message was available for each object, whereas in this model there is a certain redundance in messages. This results in a single

uniform language for Heynen’s model (ensured by the "Drastic Updates" - see 3), whereas in this model different end-languages or dialects can form, as can be seen in figure 4.3.

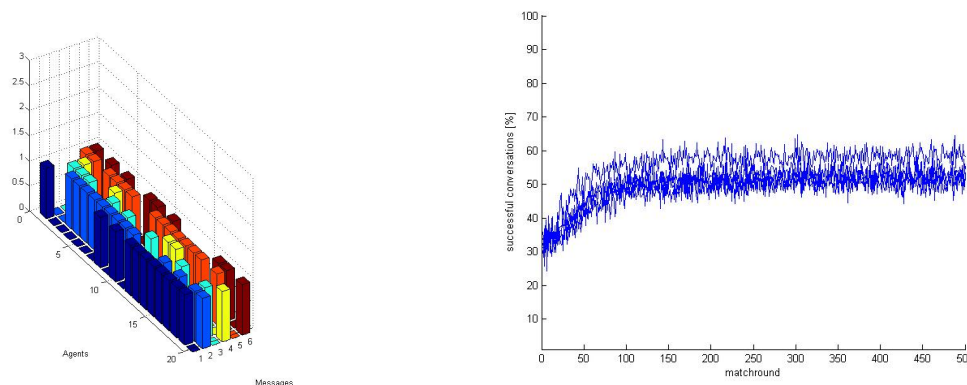


Figure 4.3 – Final Bar graph with its success rate of the 1vs1 round

As can be seen in the language graph on the left, several sets of similar but still distinct dialects are formed, with only a few cases of identical languages. The success rate stabilizes at a rather low value, between 50% and 60% , as would be expected from such a diverse mix of dialects. This situation is indicative of very early language formations over vast regions, with each agent representing a small tribe itself rather than an individual. A clear example of this are Latin languages, which all have a common root (set of possible messages) but still differ from another. And as expected, it is still relatively simple for two people who both speak latinic languages, say for example Italian and Spanish, to understand each other.

4.2.2 2vs2

In this run there are already two distinct tribes each with a inkling of a language, however, with a starting value of 2, both these languages are still quite young. Indeed, although the end languages are already slightly more grouped between blue and red, both tribes still adopt parts of the other tribe’s language to a large extent, roughly 30% .

The coherence of the formed languages is most evident in the success rate, which is distinctly higher than in the previous case, stabilizing between 60% and 70% .

4.2.3 3vs3 and 5vs5

Already at a starting value of 3, each tribe tends to keep it’s own language with very few individuals adopting elements of the other tribe, as apparent in the below graph.

Naturally this is even more pronounced with even higher starting values: with a starting value of 5, mutual influence has all but vanished.

Although the languages are already quite solid with a starting value of 3, the difference in change between situation 3 and 5 is still noteworthy at roughly 5% , so the upper threshold for language resistance has also been chosen at 5.

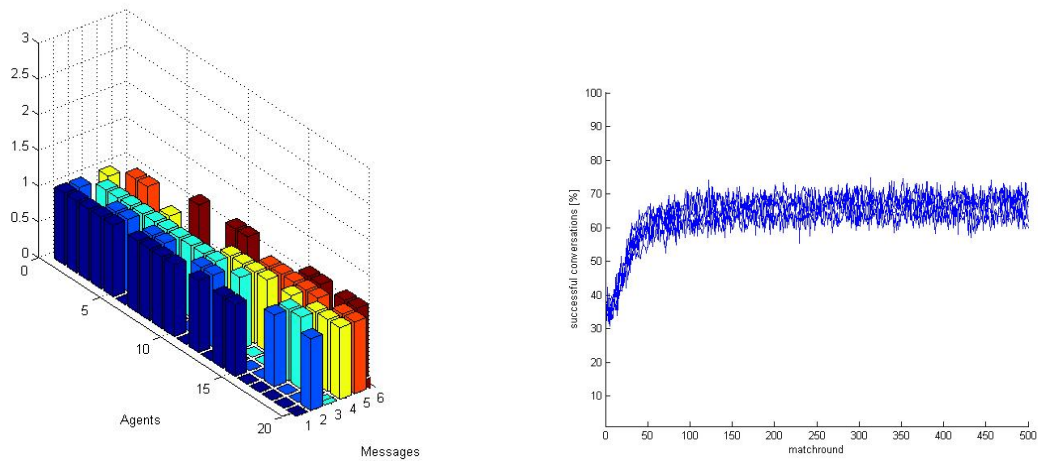


Figure 4.4 – Final Bar graph with its success rate of the 2vs2 round

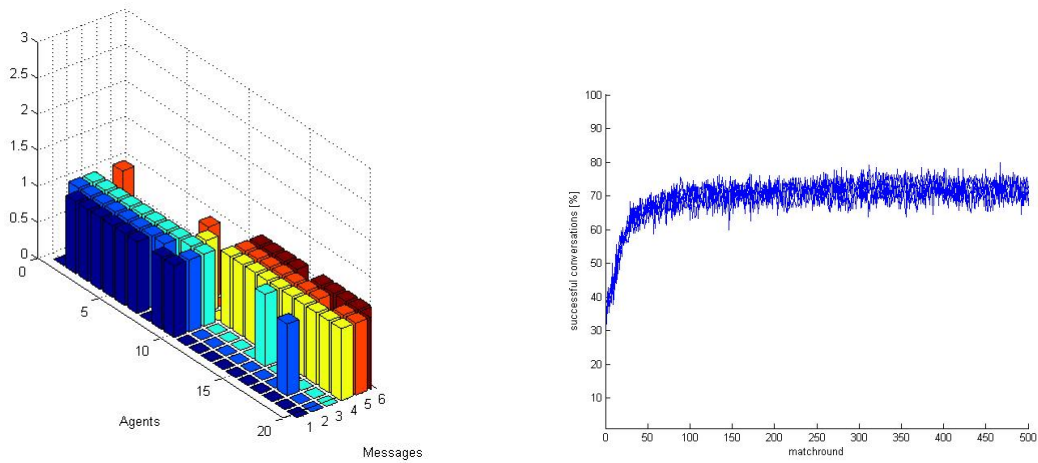


Figure 4.5 – Final Bar graph with its success rate of the 3vs3 round

4.3 Case III

In the third case study, the populations of both tribes are still the same, however the starting values or "age" of each tribe's language is different.

4.3.1 5vs1

In this run, extreme levels were set for both tribes, Blue tribe having a very developed language whereas Red tribe not having any language yet, and therefore be open to any new language. It is expected that Blue tribe incorporates Red tribe entirely and ends up simply with a much larger population.

Surprisingly figure 4.7 shows that Red tribe does not completely assume the Blue language but rather still forms it's own dialects. From the success graph it is however clear, with a success rate between 80% and 90% , that all members from both tribes are able to communicate effectively. This actually corresponds quite well to reality, i.e. in the case of regional dialects,

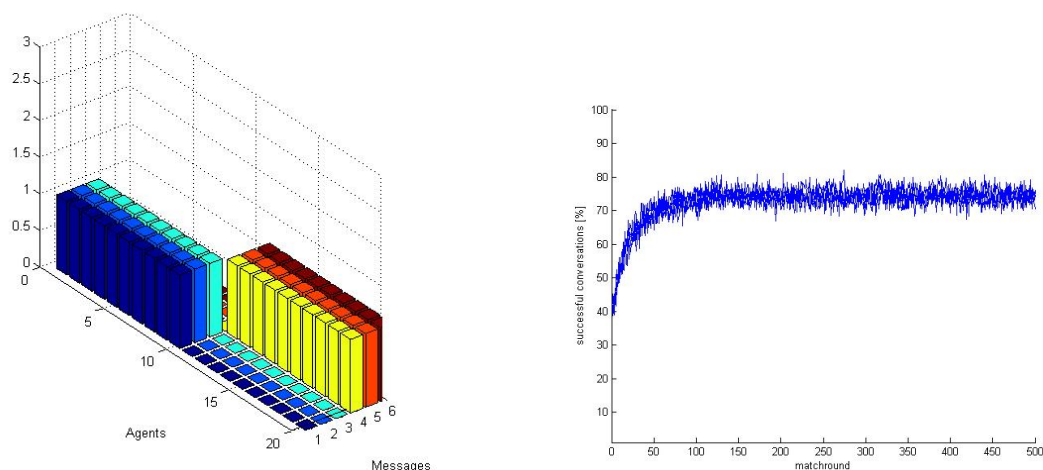


Figure 4.6 – Final Bar graph with its success rate of the 5vs5 round

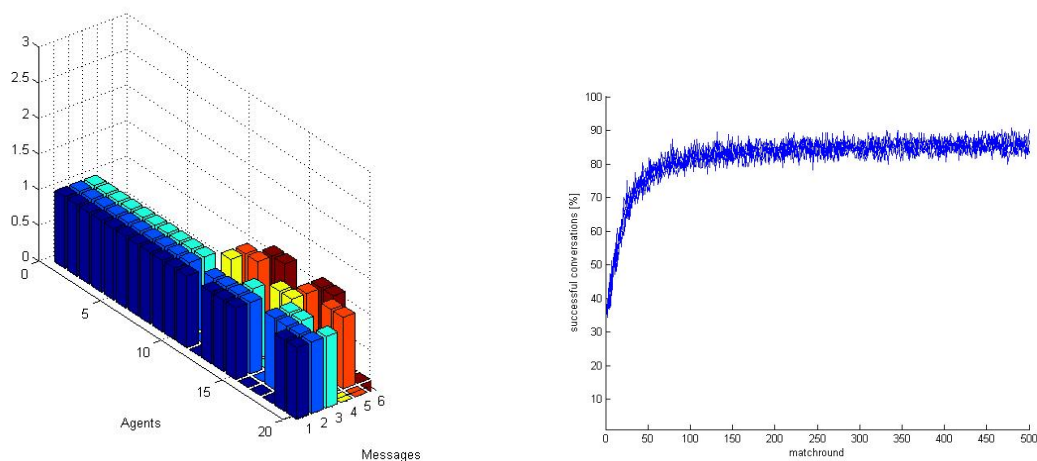


Figure 4.7 – Final Bar graph with its success rate of the 5vs1 round

where vocabulary is often somewhat different but the underlying syntax of a language is the same.

4.3.2 5vs2

In this run, the influence of a very developed language on a very young language is examined. Since, as was shown in Case I and Case II, the blue tribe is both resistant and powerful so all influence should still be uni-directional.

This time the results are as expected: it can be immediately observed that the success rate is still lower than in run 4.3.1 (see figure 4.7), where the Red tribe is in a situation to directly absorb the Blue language. However, though the Blue language is partially absorbed by Red tribe, it almost seems as if Red tribe is rebelling by keeping the Red language as much as possible, generally keeping over 50% of it's own language.

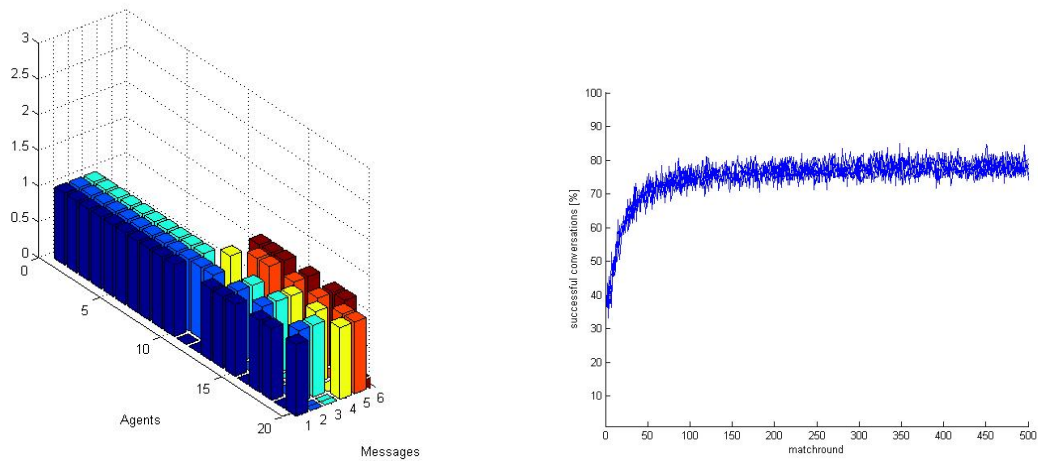


Figure 4.8 – Final Bar graph with its success rate

4.3.3 3vs2

This situation is of utmost interest, with a slight difference in language age and bi-directional influence.

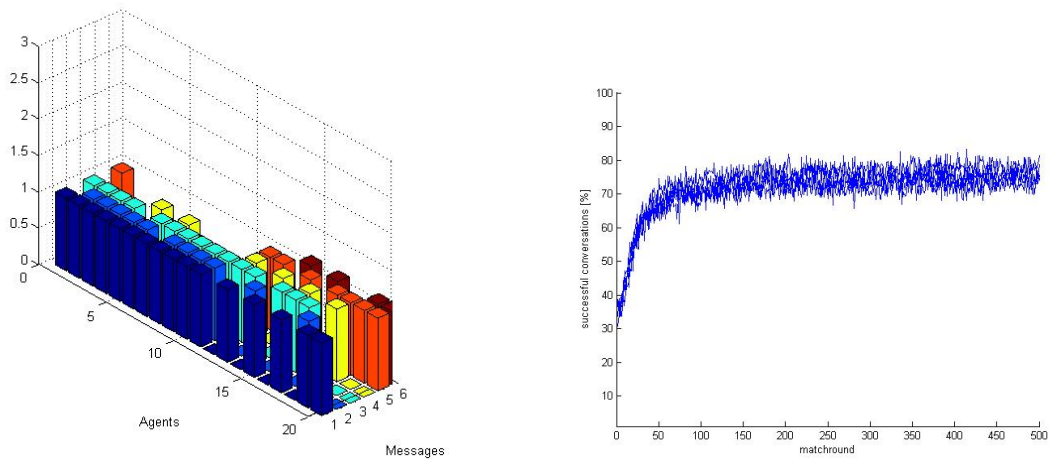


Figure 4.9 – Final Bar graph with its success rate

Figure 4.9 shows that this time both tribes are open to changes, although of course Blue tribe, with a higher starting value, is somewhat more resistant. So levels 3 and 2 are already strong and contribute to two distinct linguistic roots, however mutual learning is still possible. It is also interesting to note that the success rate has remained high albeit taking a longer than usual to stabilize completely, at just under 200 rounds, and also has a somewhat wider range.

4.4 Case IV

Until now, tribes of equal population were studied. This case now examines key situations found in the previous case studies, but adjusting the amount of agents in each tribe. This models

minorities particularly well. Of particular interest in this study are situations where the smaller tribe has a more developed language. It is of great interest to see which factor has a greater influence: population or language age. For the sake of clarity, the total population of both tribes is kept at 20.

4.4.1 3v2, Blue 5 Red 15

In this run the starting values are the same as in case 4.3.3, meaning bidirectional influence is still possible. However this time Red tribe, although having a younger language, has a much greater population, outnumbering Blue tribe three to one.

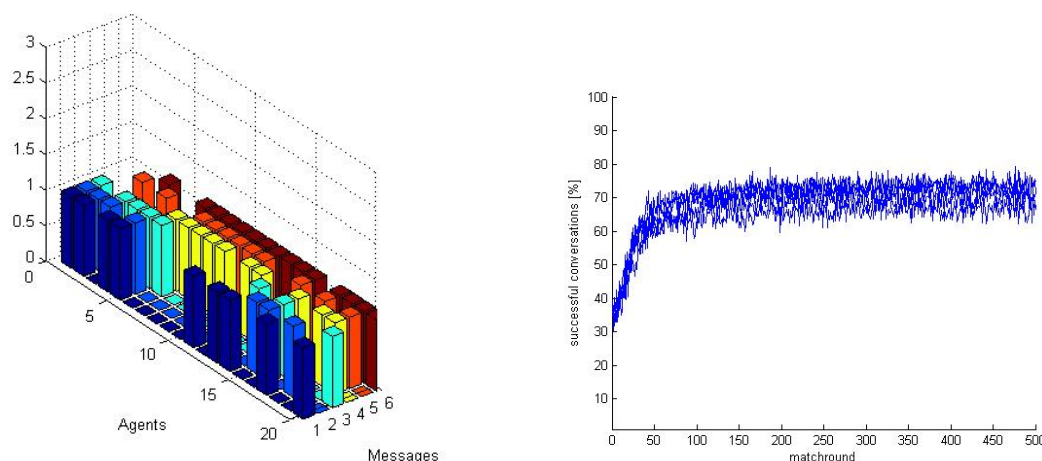


Figure 4.10 – Different number of agents with different level of the language

As can be seen, despite being heavily outnumbered, Blue tribe still maintains a large portion it’s own language, however Red tribe also displays greater resistance. In fact, the success rate stabilizes at a somewhat low value between 60% and 70% .

4.4.2 5v1, blue 1 red 19

The purpose of this run is to verify the conclusion of case 4.3.1, specifically, if an old language with a starting value of 5 really is strong enough to resist all outside influence, even in the case of being outnumbered 19 to 1! This could seem unlikely since the single member of Blue tribe will only converse with Red tribe members, and therefore foreign languages.

From figure 4.11 we can see that Red tribe has indeed evolved uniformly random dialects, just as in case (1v1), however in all cases (several simulations were run and examined) the single member of Blue tribe keeps the Blue language intact, confirming this threshold value.

4.4.3 3v2, blue 12 red 8

In the final case studied in this report there are only slight difference in both starting values and population, however Blue tribe has both the older language and the larger tribe.

Comparing figure 4.12 with case 4.3.3, the most apparent difference is the higher success rate, which can be attributed to the fact that a larger percentage of the total population speaks the

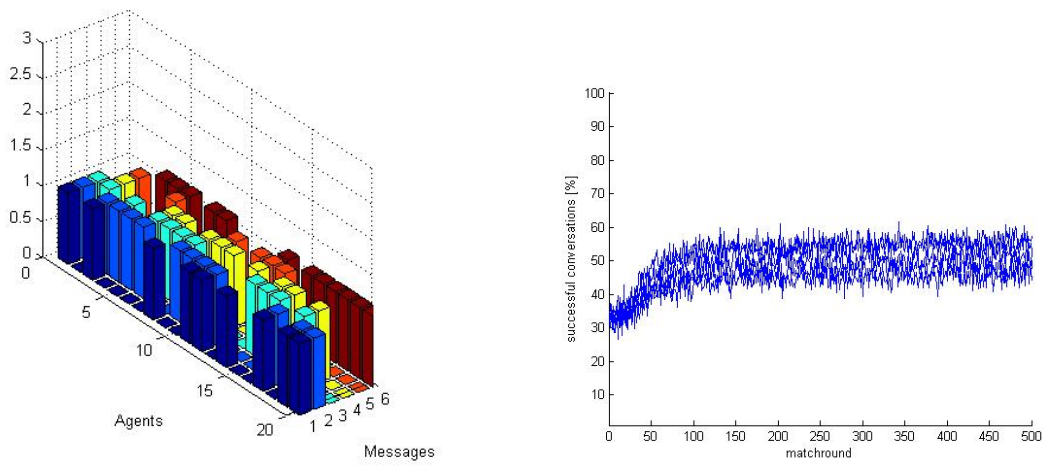


Figure 4.11 – Different number of agents with different level of the language

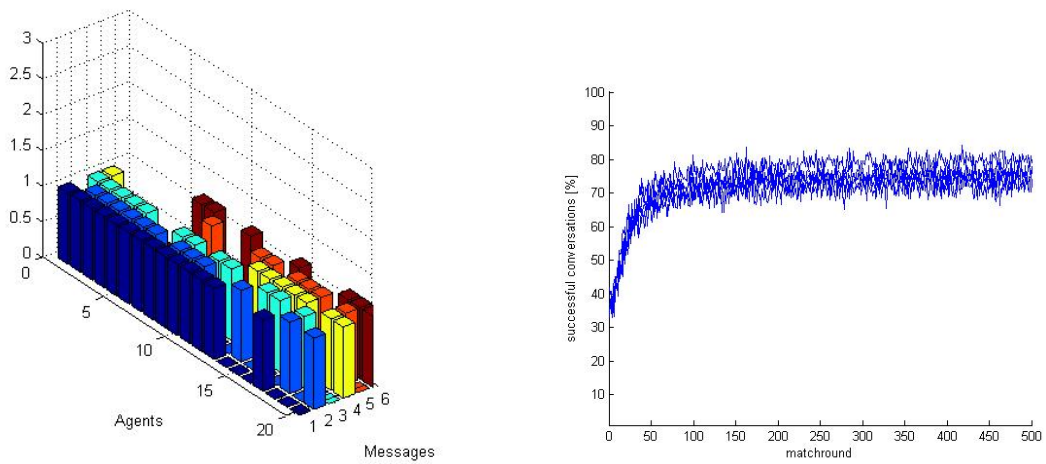


Figure 4.12 – Different number of agents with different level of the language

same language (Blue tribe). It is however very interesting to note that the influence exerted by Blue tribe on Red tribe doesn't seem to increased despite now outnumbering Red tribe three to two. This confirms the observation of case 5.5.2 that the population plays a role in a languages resistance, but not necessarily in the languages power of influence.

5 Conclusion and Outlook

5.1 Conclusion

After an exhaustive amount of simulations and analysis, several conclusions have been made. First, the age threshold for both a language's influential power as well as resistance is at a starting value of 5, relative to this model. More importantly, this means that power and resistance are proportional to the language age in the same way.

Second, and quite surprisingly, a tribe starting without its own language will always form separate languages/dialects to some extent, regardless of the influential power of neighboring tribes. Third, the size of a tribe influences the resistance of that tribe's language, but not, or only to a very small extent, the influential power. Naturally, if one considers real situations, a greater population will generally mean that a certain tribe will be spread over a larger region, and therefore influence more foreign tribes at the same time, however the rate and extent of influence is not, according to this model, changed.

5.2 Outlook

One of the limitations of this model is that each agent can only know one language, meaning for each agent, there is only one object-message link with a high value. This study has focused primarily on relatively young languages: although a language with a starting level of 5 is considered "old" in this model, it is actually still a young language. Indeed, it is most likely that for older languages, the accuracy of this model is no longer very high, since even modern languages often accept and absorb foreign words, often without even replacing existing. Therefore a very interesting next step would be to extend the model's functionality to be able to handle the concept of "synonyms", basically where each object would be allowed to have more than one corresponding message. This would be particularly helpful to understand the interactions between semi-modern languages, such as the influence of Chinese on Japanese, which despite already having a relatively well formed syntax absorbed a very large amount of Chinese.

Bibliography

- [1] Steel L, and Kaplan F, "*Bootstrapping Frowned Word Semantics*" *Linguistic Evolution through Language Acquisition: Formal and Computational Models* Ed. T. Briscoe. Cambridge University Press, 2002
- [2] Heynen M, *Model of the social emergence of lexicon* 2009
- [3] www.wikipedia.org consulted on May 20th 2010
- [4] Perfors A, *Simulated Evolution of Language: a Review of the Field* *Journal of Artificial Societies and Social Simulation* vol 5, no. 2, 2002
- [5] Selten R, Warglien M, *The emergence of simple languages in an experimental coordination game* *PNAS*, vol. 104, no. 18, 7361-7366, 2007
- [6] Gostoli U, *A Cognitively Founded Model of the Social Emergence of Lexicon* *Journal of Artificial Societies and Social Simulation*, vol. 11, no. 12, 2008