



Norwegian University of
Science and Technology

Combining Evolutionary Algorithms and Language Games to Simulate Language Emergence

Per Odlo

Master of Science in Computer Science

Submission date: June 2017

Supervisor: Björn Gambäck, IDI

Norwegian University of Science and Technology
Department of Computer Science

Per Odlo

Combining Evolutionary Algorithms and Language Games to Simulate Language Emergence

Master's Thesis in Computer Science, Spring 2017

Data and Artificial Intelligence Group
Department of Computer Science
Faculty of Information Technology and Electrical Engineering
Norwegian University of Science and Technology



Abstract

This thesis is focused on naming games and other language games and how they can be used to simulate the emergence of human languages.

One specific naming game model proposed in the literature is reviewed and examined in detail. It is successfully reimplemented with the intention of verifying the results presented in the original article. The resulting implementation serves as a starting point for the development of new naming game model.

The proposed new model incorporates elements known from evolutionary algorithms into the naming game framework. The model is implemented and used for simulations of language evolution in several scenarios. The results are discussed in relation to other naming game models and human language evolution.

The thesis is concluded with a summary of what has been achieved and an outline of possible areas for future work in the field.

Sammendrag

Denne masteroppgaven fokuserer på navneleker og andre språkspill, samt hvordan de kan brukes til å simulere måten menneskelige språk oppstår på.

Én spesifikk navnelek-modell hentet fra litteraturen blir gjennomgått og undersøkt i detalj. Den blir re-implementert, med en intensjon om å kunne verifisere resultatene fra artikkelen hvor den ble beskrevet. Den resulterende implementasjonen fungerer videre som et utgangspunkt for en ny navnelek-modell.

Den foreslåtte navnelek-modellen inkorporerer elementer kjent fra evolusjonære algoritmer inn i navnelek-rammeverket. Modellen implementeres og brukes til simuleringer av språkutvikling i flere ulike scenarioer. Resultatene fra den diskuteres i relasjon til andre navnelek-modeller og menneskelige språk sin utvikling.

Masteroppgaven avsluttes med en oppsummering av hva som har blitt oppnådd og noen idéer for fremtidig arbeid i feltet.

Preface

This Master's thesis was carried out during the 10th and final semester of the Master's degree in Computer Science at the Norwegian University of Science and Technology (NTNU), in Trondheim. The project covers 30 credits, representing the workload of one full semester.

I want to thank the Department of Computer Science (IDI) for allowing me to do my work within the two equally fascinating fields of computer science and linguistics.

I also want to thank my adviser Björn Gambäck for his valuable feedback throughout the project.

In addition Dorota Lipowska deserves a thank you, as her research with Adam Lipowski laid the foundation for most of this thesis, and also because of her positive and enthusiastic answer when being asked for permission for the re-usage of figures.

Per Odlo
Trondheim, 13th June 2017

Contents

1	Introduction	1
1.1	Goals and Research Questions	1
1.2	Research Method	2
1.3	Contributions	3
1.4	Report Structure	3
2	Background Theory	5
2.1	Linguistics	5
2.1.1	Evolutionary Linguistics	5
2.1.2	Simulating the Emergence of Natural Language	6
2.1.3	Social Structure in Simulations	8
2.1.4	Evolution in Languages	9
2.2	Machine Learning Approaches	10
2.2.1	Evolutionary Algorithms	10
2.2.2	Genetic Algorithms	11
3	Related Work	13
3.1	Naming Games	13
3.1.1	The Simple Naming Game	13
3.1.2	The Naming Game With Alignment	15
3.2	Social Structure in Experiments	16
3.2.1	Naming Games on a Weighted Network	16
3.2.2	An Evolutionary Approach to Language Games	16
4	Implementing the Naming Game Model on Adaptive Weighted Networks	19
4.1	Architecture	19
4.2	Experimental Setup	22
4.3	Results	22

Contents

5	The Evolutionary Naming Game	31
5.1	Architecture	32
5.2	Motivation and Justification	34
5.3	Experimental Setup	39
5.4	An Example	40
5.5	Initial Results	44
5.5.1	Number of Languages in Population	45
5.5.2	Popularity of Most Used Languages	48
5.5.3	Success Rate	53
5.5.4	Social Weights	57
5.5.5	Social Network Graphs	59
5.6	Impact of Parameters	70
6	Discussion	73
6.1	Evaluation of the NGAWN Implementation	73
6.2	Evaluation of the ENG	74
6.3	Comparison of the Models	76
7	Conclusion and Future Work	79
7.1	Conclusions	79
7.1.1	Research Questions	79
7.1.2	Goal	80
7.2	Future Work	81
	Bibliography	83

List of Figures

4.1	Number of languages in NGAWN, #1.	24
4.2	Number of languages in NGAWN, #2.	25
4.3	Success rate in NGAWN.	27
4.4	Number of languages in NGAWN, #3	28
4.5	Relative number of languages in NGAWN.	29
4.6	Most common language in NGAWN.	30
5.1	$L(t)$ in ENG. $c = 1$	45
5.2	$L(t)$ in ENG. $c = 2$	46
5.3	$L(t)$ in ENG. $c = 5$	47
5.4	$N_i(t)$ in ENG, #1.	48
5.5	$N_i(t)$ in ENG, #2.	49
5.6	$N_i(t)$ in ENG, #3.	49
5.7	$N_i(t)$ in ENG, #4.	50
5.8	$N_i(t)$ in ENG, #5.	50
5.9	$N_i(t)$ in ENG, #6.	51
5.10	$N_i(t)$ in ENG, #7.	51
5.11	$N_i(t)$ in ENG, #8.	52
5.12	$N_i(t)$ in ENG, #9.	52
5.13	$N_i(t)$ in ENG, #10.	53
5.14	$s(t)$ in ENG. $c = 1$	54
5.15	$s(t)$ in ENG. $c = 2$	55
5.16	$s(t)$ in ENG. $c = 5$	56
5.17	Social weights in ENG, $c = 1$	57
5.18	Social weights in ENG, $c = 2$	58
5.19	Social weights in ENG, $c = 5$	58
5.20	Network graph #1 for ENG.	60
5.21	Network graph #2 for ENG.	61
5.22	Network graph #3 for ENG.	62
5.23	Network graph #4 for ENG.	63

List of Figures

5.24	Network graph #5 for ENG.	64
5.25	Network graph #6 for ENG.	65
5.26	Network graph #7 for ENG.	66
5.27	Network graph #8 for ENG.	67
5.28	Network graph #9 for ENG.	68
5.29	Network graph #10 for ENG.	69

List of Tables

4.1	Explanation of the parameters in Lipowska and Lipowski's experiments.	23
5.1	Explanation of the parameters in the ENG experiments.	40
5.2	The configurations used in the ENG experiments.	41
5.3	Figure reference for Section 5.5.	41
5.4	Data for a selection of dialogs in ENG, generation 1.	42
5.5	Data for a selection of dialogs in ENG, generation 3.	42
5.6	Data for a selection of dialogs in ENG, generation 20.	43
5.7	Data for a selection of dialogs in ENG, generation 100.	43
5.8	Data for a selection of generations in ENG.	44
5.9	Number of languages (L) for various values of N	70
5.10	Number of languages (L) for various values of ε	71
5.11	Number of languages (L) for various values of ρ	71
6.1	Effect of parameters in ENG.	75

Acronyms

EA Evolutionary Algorithms. 5, 10, 11, 17, 79

ENG The Evolutionary Naming Game. 3, 31, 34, 39, 74–76, 79–81

GA Genetic Algorithm. 10, 11

NGAWN The Naming Game on Adaptive Weighted Networks. 2, 3, 16, 19, 32–34, 37, 39, 73, 74, 76, 79–81

1 Introduction

The field of linguistics is an interesting one — among many others it deals with questions such as “How did human languages come to be?” and “What are the main forces behind their evolution?”. Where these research questions meet traditional computer science, we find computational linguistics.

In computational linguistics the main goal is still the same — revealing facts about the languages used by humans. But while traditional linguists do this through thorough study of today’s languages and historical records, researchers within computational linguistics try to *simulate* their way to the conclusions. A more precise explanation on the difference between the subbranches of linguistics is offered in Section 2.1.

The thesis will look at what has been done within the field of computational linguistics, and it will delve into one model in particular, later proposing an extending of the same model.

This chapter provides a brief introduction to the Master’s thesis. It states the goal of the project and the research questions, it briefly presents the research method, it lists up the major contributions of the thesis, and it gives an overview of the general report structure.

1.1 Goals and Research Questions

This section presents the main goal of the project as a whole. It is then further elaborated through the partitioning into a number of atomic research questions.

Goal *Exploring in which ways an evolutionary extension of a naming game model can be used as a simplified model of how languages evolve over time.*

1 Introduction

The naming game model, which is to be explained later, serves as a backbone for the entire thesis. It is already known that various naming game produce an emerging language within a population. In this thesis the intention is to implement a new model incorporating evolutionary aspects. It will then be held against the evolutionary process of human languages to determine to which extent similarities can be found.

Research question 1 *Is The Naming Game on Adaptive Weighted Networks (NGAWN), proposed by Lipowska and Lipowski (2012), verifiable and reproducible?*

Lipowska and Lipowski (2012)'s NGAWN is to be examined. An attempt at a verification of the model will be made by implementing it based on the authors' description and comparing the results.

Research question 2 *Can an evolutionary model based on NGAWN be constructed?*

With NGAWN as a basis, engineering a new model incorporating various evolutionary elements will be attempted.

Research question 3 *What can be achieved in terms of realistic simulations with an evolutionary model based on NGAWN?*

Then the thesis seeks to find out what the results of running simulations on the model are, and evaluate the model based on the results. Human language evolution should be looked at when the model is evaluated.

1.2 Research Method

The thesis will search to answer the research questions in a mostly practical approach. First it closely examines a proposed naming game model through literature study and implementation, before using that implementation as a basis for a more complex model. The results of simulations are then studied and compared to known facts about human language evolution. More details about the experiments can be found in Sections 4.2 and 5.3.

1.3 Contributions

The main contributions to the field from this thesis are:

- A verification of the NGAWN model proposed by Lipowska and Lipowski (2012).
- A new model called The Evolutionary Naming Game (ENG) and a justification of its setup.
- Extensive experimental results for ENG and an evaluation of it.

1.4 Report Structure

After the **Introduction**, the report continues with some **Background Theory** (Chapter 2). That chapter presents theory relevant to the project.

In **Related Work** (Chapter 3), research papers this project is based upon are described. The goal of that chapter is to establish what is the current state-of-the-art within the field.

Implementing the Naming Game Model on Adaptive Weighted Networks (Chapter 4) is the first of two chapters covering implementation. In this chapter one particular model from the literature is examined and implemented, and the results are commented.

The Evolutionary Naming Game (Chapter 5) covers a new model made on the basis of the one from Chapter 4. A description of the model is accompanied by the setup of experiments and results thereof.

Discussion (Chapter 6) discusses what has been achieved and how it relates to existing literature.

Finally, **Conclusion and Future Work** (Chapter 7) evaluates the thesis in relation to the research questions and goal stated in Section 1.1. It also looks forward and presents some areas of possible further work.

2 Background Theory

In this chapter background theory which is relevant for understanding the context of the related work, the experiments and the discussion parts of this report, namely Chapters 3, 4, 5, 6, and 7, is presented.

A brief introduction to linguistics, with a focus on evolutionary linguistics, computational linguistics and the combination of these two, will be provided. Some relevant theories will be presented, and it is described how one can simulate phenomena such as the emergence of language. The chapter concludes by providing some background material on machine learning approaches, especially Evolutionary Algorithms (EA).

2.1 Linguistics

Human language — a powerful device vital to the success of our species — remains one of the unsolved mysteries of science (Knight et al., 2000). Linguistics is the study of how to solve this mystery.

Linguistics is a broad field, seeing contributions from disciplines as (seemingly) unrelated as anthropology, neurobiology, psychology and many others (Steels, 2011). This thesis will almost exclusively concentrate on the contributions seen from the subbranches of computer science, although they obviously intervene with discoveries and theories known from other disciplines. Furthermore, while linguistics deals with phonology (the sounds in language), morphology (words and their relationships), etymology (the history of words) and an extensive list of other phenomena, this thesis will take part in the *evolutionary linguists'* search for the origin of language.

2.1.1 Evolutionary Linguistics

It seems clear that there are evolutionary forces driving the development of language, and that these forces share traits with forces present in other

2 Background Theory

areas, e.g., the evolution of species, i.e., Darwinian evolution. However, answers to what shapes the evolution of languages are still not clear, and David Premack’s 1986 quote “Human language is an embarrassment for evolutionary theory” can still be regarded as true (Steels, 2011).

Furthermore, language does not evolve in a vacuum. According to Steels it is an outcome of three interconnected processes: Biological evolution, cultural evolution, and social evolution. While the first one establishes the neurobiological necessities for spoken language, the two others shape the use and form of language within a population (Steels, 2011).

A central concept in the theory of language learning is the *Baldwin effect*, named after the American psychologist James Mark Baldwin (Schilhab et al., 2012).

According to Darwinian evolution, learned traits cannot be passed to the next generation — only genetic attributes can. This is a widely accepted scientific fact. However, according to Baldwin (1896), cultural learning, e.g., language, can guide human evolution. Specifically, this means that if having a cultural-specific trait in one’s genes will be an advantage for a species, evolution will be steered towards these genetics. This means that skills that originally had to be learned by each individual can be encoded in the genes, saving the culture from doing said teaching. It is worth noting that this principle does not contradict Darwinian evolution in any way. On the contrary, it can be viewed as an integrated part of Darwin’s theory.

2.1.2 Simulating the Emergence of Natural Language

To investigate the underlying processes of human language, many researchers have worked on various computer simulations of language acquisition, language development, and similar phenomena in populations. These are interesting because of their experimental nature — they can provide us with insight it would be difficult or even impossible to acquire through more traditional research methods (Nolfi and Mirolli, 2010).

It is, however, challenging to design these simulations in such a way that they represent a somewhat realistic setting for communication, interaction, cognitive capacities, and the dynamics in a population. One reason for this is that human culture and the human brain are too complex to be modeled in a meaningful way within the scope of such simulations. Another reason

is that many of the mechanisms and principles that make the basis of human language and the context in which it is used, are largely unknown; e.g., scientists do not fully understand every single aspect of the brain or even the parts that are concerned with language or communication.

Since realistic simulations are out of the question, linguists and computer scientists create *language games* that mimic certain aspects of human communication. In this kind of simulation one tries to look at communication between agents in a (usually) heavily restricted world following some rules. A language game is called so because it treats the participants — speaker and hearer — as players in a game where successful communication is the goal and certain restrictions in the simulation environment are the game rules (Routledge and Chapman, 2009). The term was coined by the Austrian-British philosopher Ludwig Wittgenstein in the early 1950s.

Language games consist of a *population of individuals* (the agents who communicate), a *context* (some environment for the communication to take place in), and a *communicative purpose* (a goal) (Steels, 2012). What such a goal can be naturally varies between different games, but Wittgenstein (1953) used as an example a builder (of a house or similar) that needs to communicate with his assistant, who will be handing stones to the builder.

One of the most common — and also one of the simplest — types of language games, is the naming game. It consists of a world populated by a number of agents alongside a number of objects. The agents' purpose is to agree on names for the objects. This can be done in a variety of ways — ranging from the trivial to the complex.

A simple naming game, as the one described in Steels and Loetzsch (2012), works in the following way: A number of agents with initially empty vocabularies inhabit a world with a number of unique objects. For each dialog in the game, a speaker in the population will try to draw a hearer's attention to a specific object by uttering a word — either one already in his vocabulary or one he invents on the fly. If the hearer is able to correctly identify the object, the dialog is a success. The goal of the game is to over time develop the same vocabularies across agents, and thus have a high number of dialog successes. This game is explained in detail in Sections 3.1.1 and 3.1.2.

A game as the one proposed by Steels and Loetzsch above will (if suc-

2 Background Theory

cessful) develop a set of words and meanings. However, it does not include any behavior complex enough to create any relation between the words — there is no grammar. While this does not sound to faithfully represent any version of the complex human languages we know today, it resembles the *protolanguage* described by Bickerton (1990). He writes that the first steps towards modern languages — the protolanguage — were taken somewhere between 500,000 and 1.5 million years ago, and that such a protolanguage is characterized by a lexicon containing a number of concepts, but with no grammatical elements and no syntax.

However, more complex language games exist, and some of them address this lack of syntactic structure. For example, van Trijp (2012) looks at a language game meant for developing case structure among agents, which is done with quite some success.

2.1.3 Social Structure in Simulations

An important aspect of human culture is the social relations — “no man is an island”, as written by the English poet John Donne in 1624. These relations form social networks, a phenomenon that has been studied by many different scientists in many different contexts (Costa et al., 2011). With the rise of modern technology such as social media, these networks continue being an important area of research.

Of course, when simulating language, social relations should not be ignored, or at least not without a valid reason. It is therefore no surprise that a great portion of today’s research on agent communications has a focus on the networks the communications take place in context the context of. Part of the background for this is that the assumption that any two agents communicating is equally probable, as in the simple naming game (Steels and Loetzsch, 2012), is highly unrealistic (Baronchelli et al., 2006).

There are countless ways to arrange the agent in a social network. Gong et al. (2004) choose to to implement a model with numerical weights between the agents, with the weights being manipulated following successful and unsuccessful communication attempts. A similar social network also serves as the foundation of Lipowska and Lipowski (2012), further discussed in Section 3.2.1 and Chapter 4.

Social relations do not need to be restricted to a (weighted) network.

One can also introduce social roles, rules governing communication within the network, etc. One implementation using some of these elements can be found in Lekvam (2014).

2.1.4 Evolution in Languages

It is an open question whether all the thousands of languages existing today share the same root or not. But as Bickerton (2007) point out, it does not necessarily matter. In any case, languages have evolved, split and died out for a long time, possibly several million years. As should be obvious, the origins of languages is not something one can hope to obtain a good understanding of anytime soon. It is easier, and perhaps just as interesting, to look at the forces and processes behind language evolution in a smaller time-frame. In a way, one can say that linguists traditionally work their way backwards: They start with what is known (the present languages) and make hypotheses for earlier languages based on evidence and known rules. When using computer simulations, on the other hand, the starting point is the opposite: One starts with what is given (the absence of any language) and simulate one's way forward based on models and assumptions.

Nevertheless some linguists have speculated about the origin of languages. Hurford (2003) writes that the proto-language — the imagined first language among humans — very well may have had a much simpler structure than today's languages. For example he writes that there might not have been any proper names, fewer words with less specific meanings, less differentiation between syntactic classes such as verb and noun, and fewer grammatical elements.

This view fits into what is known as the *grammaticalization theory*: The theory that grammatical structures originates as words without a grammatical function that are used in a specific role often enough to become an integrated part of the language's grammar. Thus, the starting point for languages seems to be a non-syntactic organization, i.e., more of a collection of words than a syntactic structure.

2.2 Machine Learning Approaches

Traditionally, algorithms used in various computer science branches have been carefully engineered to represent how the programmers think solutions should be found. This has led to rule-based systems. This approach has been common within the field of Artificial Intelligence too — with algorithms such as Minimax or A*. However, another trend has emerged: Using *Machine Learning* methods, where the algorithm is not specified entirely by the programmer, and it can change (and hopefully improve) over time.

2.2.1 Evolutionary Algorithms

A current trend in Machine Learning is drawing inspiration from biological processes in nature. A prime example of this is the use of *EA*.

Methods used by EA are reproduction, mutation, and natural selection, among others.

Types of EA include genetic programming and evolutionary programming, where the solutions to be found are computer programs set out to solve specific computational problems. The Genetic Algorithm (GA) is examined in GA.

While the mechanisms prevalent in GA and EA in general could be explained in great detail, that will not be done in this thesis, as the work done in this thesis do not to any particular degree integrate specific techniques know from other EA. The curious reader is recommended to have a look at Ashlock (2006) for an introduction to the field.

Whether natural selection plays a role in language adaption and evolution is not entirely uncontroversial. Linguists debating this question can generally be split into two groups: The *adaptionists* who argue that natural selection as first described by Darwin is the only way to explain language, and that evolutionary biology is a highly relevant context to study language within. On the other hand, the *non-adaptionists* argue that language evolution rely on “spandrels” — that language skills did not evolve as a fitness advantage to its users as much as it evolved as a side-effect of other skills (Szathmáry, 2010).

Using EA on linguistic problems and hoping to achieve something in terms of an explanation of human language, inevitably seems to position

oneself among the adaptationists. However, it fits neatly the trend of relying on *agent-based modeling* to study social dynamics, a trend that is also prevalent in computational linguistics. The goal of such models is to move the problem of emergence from the micro level to the macro level, constructing simple agents modeling a complex phenomenon (Loreto, 2010).

2.2.2 Genetic Algorithms

One of the most well-known EA is the GA. These algorithms operate with a number of possible problem solutions as individuals in a population. The goal for the GA is to produce the best solutions, judged by some fitness measure set by the programmer. To achieve this, the principles of Darwinian evolution are applied: The fittest solutions reproduce, resulting in individuals with similar traits as themselves, though random mutations also occur. When applying this approach to rather large populations over a span of several generations, impressive and sometimes surprising solutions can be found.

A typical GA consists of the following steps:

1. **Initialization:** First a number of random solutions (binary strings or whatever fits the problem at hand) is generated.
2. **Selection:** For every generation, a number of solutions/agents are selected to reproduce. Many different selection strategies exist, but they all use the *fitness function* to assert a probability of choosing a specific agent. The fitness function is a problem-specific measure of how good a specific solution is. If an exact fitness function is hard to find, simulations can be used.
3. **Genetic operators:** When two agents are chosen to produce an offspring, the offspring should have a mix of the genes of the two parents. This happens through a two-step process:
 - a) **Crossover:** The two solutions are combined by selecting a random crossover-point. For example, if the two parent solutions are “0000” and “1111”, and the crossover-point is chosen to be between the second and third digit, the new agent can get the solution “0011”.

2 Background Theory

- b) **Mutation:** In order find new solutions it is often not enough to recombine the existing ones. Therefore there is often a given probability for a mutation. Should this happen, is the most common strategy to flip a single bit in the representation of the solution (in case of a binary string).
- 4. **New generation:** The newly generated agents take over for the existing ones, and one goes back to step 2. Whether some old agents survive and continue to be in the pool or not, vary between implementations.
- 5. **Termination:** If a good enough solution is found, or a pre-defined number of generations has passed, the algorithm is terminated.

3 Related Work

In this chapter a collection of related work within the field will be presented. This work forms the basis for the research and experiments presented later in the thesis.

3.1 Naming Games

One of the most common language games is the naming game. The simplest form of it is described in Section 3.1.1. More complex variants will be further explored later.

3.1.1 The Simple Naming Game

In Steels and Loetzsch (2012), Luc Steels and Martin Loetzsch describe a simple naming game which has already been outlined in Section 2.1.2. Here a more thorough description will follow.

A round in the naming game consists of a speaker trying to draw a hearer's attention to a specific object by referring to it in such a manner that the hearer is able to correctly identify it. Specifically, these points constitute a round:

1. A speaker, a hearer and an object are selected.
2. The speaker selects the name for this object from his vocabulary and then utters it.
3. The hearer looks of this name in his vocabulary and identifies which (if any) object is associated with it.
4. The hearer signals to the speaker which object he has identified.

3 Related Work

5. The speaker checks whether this is the correct object. If it is, the game is a success, and if not, it is a failure.
6. The hearer gets to know the outcome of the game.

The interesting parts in this procedure are the ones involving the agents' vocabularies. Which words reside there, and why? Steels and Loetzsch point out that the game is most interesting when the agents start without any prior vocabulary and have to invent words for the objects as the game goes on. This can be carried out using a few simple rules:

- If the speaker has no name for the object, he invents a name on the fly.
- If the hearer does not know the name used by the speaker, the game is a failure, and the hearer inserts this word in his vocabulary.
- If the hearer signals the wrong object (because the word is used differently in his vocabulary), the game is a failure, and he associates the word with the object the speaker meant in his vocabulary.

Note that both *homonyms* and *synonyms* can exist — meaning that in an agent's vocabulary, a word can be associated with several objects and an object can be associated with several words, respectively. If a new association is made, previous associations of this word or object are not deleted. As a consequence, the game can result in a failure even when the correct word-object association is in the hearer's vocabulary, as he is forced to pick an object among several possible. Another problem with this straight-forward approach is that it creates far more synonyms than desirable. In a way, one can say that the game gives each agent the knowledge of all other agents' languages instead of creating something that converges into a single, common language.

Steels and Loetzsch run experiments where they tried out these simple rules on a small population of ten individuals, placed in a world with five different objects. The results after five independent runs show that the number of successful dialogs steadily rises from zero to all within about 500 dialogs per agent. The authors point out that this means that the communication systems has the required expressive adequacy — the

evolved language is able to express everything in the world with absolute precision. However, they also point out that the resulting vocabulary is not exactly optimal: on average, each object has nearly five synonyms.

3.1.2 The Naming Game With Alignment

The authors Steels and Loetzsch (2012) describe a mechanism designed to avoid the situation with a too large vocabulary, described in the previous section, which they call *alignment*. Specifically, one particular method known as the *lateral inhibition strategy* is explained. Using this strategy, all word-object associations in the vocabulary of an agent are given a numerical score, initially some σ_{init} . Any time a game results in a success, both the speaker and the hearer increases the score of the association between the word and the object used in the game by a fixed constant $\delta_{success}$. After a failed game, the speaker and the hearer both decrease the score of the used association by a corresponding $\delta_{failure}$. Associations that reach a score of 0 will reach an inactive state and will not be used again unless their scores increase again, which can happen if another agent uses that word.

This approach causes the (active) vocabularies of the agents to decrease in size and eventually stabilize in a situation with no synonyms or homonyms.

The experiments conducted using this strategy yield the following results: The communicative successes need substantially less time to reach 100 % with this approach. In addition it can be seen that the size of the vocabulary actually decreases after the initial phase. It eventually stabilizes at one active name per object, the simulation thus having created an optimal vocabulary. The authors conclude that this was indeed the emergent behavior they wanted to achieve, and that alignment based on the outcome of communication triggered the necessary self-organization.

What the authors call “the Grounded Naming Game” — as opposed to the “Non-Grounded” variant explained above — utilizes physical, embodied agents and objects, which adds complexity to the task. That path will not be followed in this thesis. However, it deals with a few other interesting attributes, among others introducing the need to map from a concept in the mind of an agent to the correct physical object.

3.2 Social Structure in Experiments

As argued in Section 2.1.3, there is good reason to incorporate social structure in language experiments. A few articles doing so is presented in this section.

3.2.1 Naming Games on a Weighted Network

Lipowska and Lipowski (2012) present a naming game based on social relations implemented as weights between individual agents, called the The Naming Game on Adaptive Weighted Networks (NGAWN).

The naming game outlined in (Steels and Loetzsch, 2012) is extended with a numerical weight for each agent-pair. Successful communications increase this weight, while unsuccessful ones decrease it. Additionally, in each dialog the speaker has a bias towards picking a hearer with which it has a high weight value. This simulates the emergence of an internal social structure in the population.

Lipowska and Lipowski find that while a clear social structure is defined, the number of languages steadily decreases and even reaches one — i.e., all agents share the same language — within some configurations. However, they are also able to produce configurations where the populations reach a stable state of several languages co-existing, akin to the way the human population is split into a number of sub-populations with their own language.

3.2.2 An Evolutionary Approach to Language Games

In Lipowski and Lipowska (2008), Adam Lipowski and Dorota Lipowska examine a simulation of a naming game model for a set of communicating agents.

On the surface, the game lies very close to the different naming games already discussed in this chapter, with the goal of establishing a common vocabulary through dialogs between a speaker and a hearer. The model also incorporates weights for each word in an agent's vocabulary, similar to the alignment in Section 3.1.2.

What separates Lipowski and Lipowska's approach from the plain naming games, though, is that they simultaneously evolve the population with

3.2 Social Structure in Experiments

breeding, mutation, and death. That is, they place the language game in the context of some sort of Evolutionary Algorithms (EA).

The communicative success of an agent does not determine for how long it survives or whom it will breed with. Its vocabulary does, however, play a role when the agent has an offspring: The offspring inherits the agent's highest scoring word. (At this point it should probably be mentioned that Lipowski and Lipowska tried to establish a vocabulary for one object only, though they argue that their approach can be used for larger environments as well.)

The genetics, however, affect the communication simulation. Each agent has a learning ability encoded in its genes, which is subject for inheritance and mutation when the agent is born. This learning ability decides how the weights of words will be modified following a dialog.

The experiments conducted by the authors show that the population quickly achieves a high learning rate and a 100 % communicative success. They also find that when allowing many dialogs and little evolution in a short span of time, the success rate goes up, followed more slowly by the learning rate. This means that fast cultural changes guide evolution, a phenomenon known as the Baldwin effect, explained in Section 2.1.1.

The model is further analyzed in (Lipowska, 2011). Here the author points out that the speed of cultural changes might be a factor affecting the evolution of language.

Lekvam et al. (2014) were not able to reproduce all the results from (Lipowska, 2011). They write that the model was in fact not as robust as Lipowska argued, and that the parameters need to be monitored carefully to be able to maintain a stable environment. They were, however, able to reproduce some of the results, but not with the same parameters as were used originally.

4 Implementing the Naming Game Model on Adaptive Weighted Networks

The very core of this thesis is the execution of various language experiments. This chapter covers the first half of those experiments, which then forms a basis for other experiments in Chapter 5.

The Naming Game on Adaptive Weighted Networks (NGAWN), proposed by Lipowska and Lipowski (2012), is implemented. The chapter begins with a detailed description of the implementation, and follows up with results from various experiments. Those are compared to the original results of the Lipowska and Lipowski article. Some concluding remarks about the significance of said results are left to the general discussion (Chapter 7).

4.1 Architecture

Although an outline of NGAWN is presented in Section 3.2.1, a more detailed breakdown of the algorithm, including pseudo-code, is offered in this section.

$$P(b) = \frac{w(a, b) + \varepsilon}{\sum_x (w(a, x) + \varepsilon)} \quad (4.1)$$

Formula 5.1 shows the probability of speaker a choosing hearer b based on the weights w ($\frac{\text{number of dialogs}}{\text{number of successes}}$).

Algorithm 1 is the starting point of the simulation of a dialog. The other algorithms presented are called by the main algorithm.

Algorithm 1 NGAWN, general simulation

Input: *agents*, a set of agents with vocabularies

Input: *dialogs*, an array with the number of performed dialogs for each agent-pair

Input: *successes*, an array with the number of successful dialogs for each agent-pair

Returns: a boolean saying if the game was a success or not

```
1: speaker ← randomly drawn agent from agents
2: hearer ← choose_hearer(speaker, agents, dialogs, successes)
3: word ← speak(speaker)
4: success ← listen(hearer, word)
5: if success then
6:   speaker.retain_only(word)
7:   hearer.retain_only(word)
8: else
9:   hearer.vocabulary.add_word(word)
10: end if
11: dialogs[speaker][hearer] ← dialogs[speaker][hearer] + 1
12: if success then
13:   successes[speaker][hearer] ← successes[speaker][hearer] + 1
14: end if
15: return success
```

Algorithm 2 The **choose_hearer** function of NGAWN

Input: *speaker*, a chosen speaker for the dialog

Input: *agents*, a set of agents with vocabularies

Input: *dialogs*, an array with the number of performed dialogs for each agent-pair

Input: *successes*, an array with the number of successful dialogs for each agent-pair

Returns: a hearer for the dialog from *agents*

```

1: weights  $\leftarrow$  sum of successes[speaker][agent]/dialogs[speaker][agent]
   for all agent in agents
2: random  $\leftarrow$  a random real number between 0 and weights
3: temp  $\leftarrow$  0
4: for all agent  $\neq$  speaker in agents do
5:   success_ratio = successes[speaker][agent]/dialogs[speaker][agent]
6:   temp  $\leftarrow$  temp + success_ratio +  $\varepsilon$ 
7:   if temp > random then
8:     return agent
9:   end if
10: end for

```

Algorithm 3 The **speak** function of NGAWN

Input: *speaker*, an agent to utter a word

Returns: a word to be used in the dialog

```

1: if speaker.vocabulary is empty then
2:    $\#$  Add randomly generated word
3:   speaker.vocabulary.add_word(new_word)
4: end if
5: return randomly chosen word from speaker.vocabulary

```

Algorithm 4 The **listen** function of NGAWN

Input: *hearer*, an agent to hear a word

Input: *word*, an uttered word

Returns: a boolean saying if the hearer understood the word or not

```

1: if hearer.vocabulary.contains(word) then
2:   return true
3: else
4:   return false
5: end if

```

4.2 Experimental Setup

The main goal for the experiments conducted in this chapter is to reproduce many enough of Lipowska and Lipowski (2012)'s results to be able to verify them.

In order to do that, the same configurations, including number of agents (N) and parameter for selecting a hearer (ε) has been used. In some cases the number of generations (t) has been reduced, but not where any important changes happen in the later stages of the original results.

The results presented in Section 4.3 are:

- The number of languages L , calculated for $\varepsilon = 10^{-4}$.
- The number of languages L , calculated for $\varepsilon = 10^{-5}$.
- The success rate s , calculated for $N\varepsilon^2 = 10^{-5}$.
- The number of languages L , calculated for $N\varepsilon^2 = 10^{-5}$.
- The number of languages relative to the population size, L/N , calculated for $N\varepsilon^2 = 10^{-5}$.
- The share of agents using the most common language, N_d/N , calculated for $N\varepsilon^2 = 10^{-5}$.

4.3 Results

As explained above, in Lipowska and Lipowski (2012)'s naming game, most agents initially invent a likely unique word. Therefore we say that

L	The number of languages in the population
N	The number of agents in the population
N_d	The number of agents using the most popular word
t	The number of generations simulated
ε	Parameter used when speaker selects a hearer ^a
s	The communicative success rate ^b

^aA larger ε means more variance, i.e., higher chance of choosing hearers with which the speaker has an unsuccessful communicative history (low relational weight). For the formula, see Formula 4.1.

^bThe share of successful dialogs.

Table 4.1: Explanation of the parameters in Lipowska and Lipowski’s experiments.

each of them have their own language. As the game progresses, some agents discard their own language and adapt that of another agent, so that the total number of languages is reduced. In the first experiments with this game, it is examined how the number of languages L present in the population develops over time t , for different population sizes N . One time-step represents one generation in the simulation. Each generation involves N dialogs with randomly chosen agents, meaning that each agents will on average participate in two dialogs per generation, as either speaker or hearer.

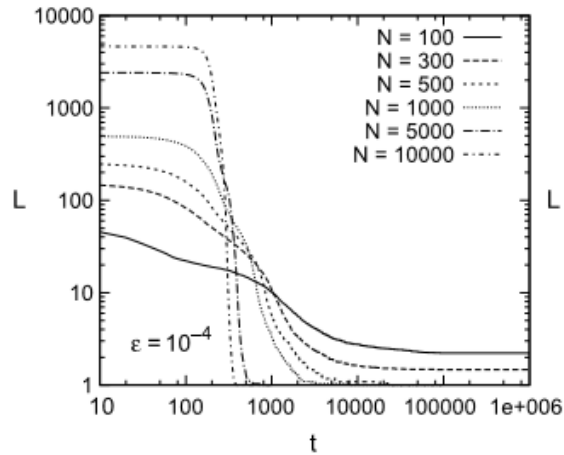
All experimental results presented are based on the average of ten independent runs, which should be enough to be able to obtain results with a relatively small margin of error. Lipowska and Lipowski also present the average of several runs in their graphs, but the exact number is not specified in their article.

The implementations have been as faithful to the descriptions in Lipowska and Lipowski (2012) as possible, in the way interpreted and shown in Section 4.1. The experiments have been conducted using Java¹, a relatively fast programming language allowing for the simulation a large amount of games in a short time. Below the individual results obtained are discussed in relation to the original research.

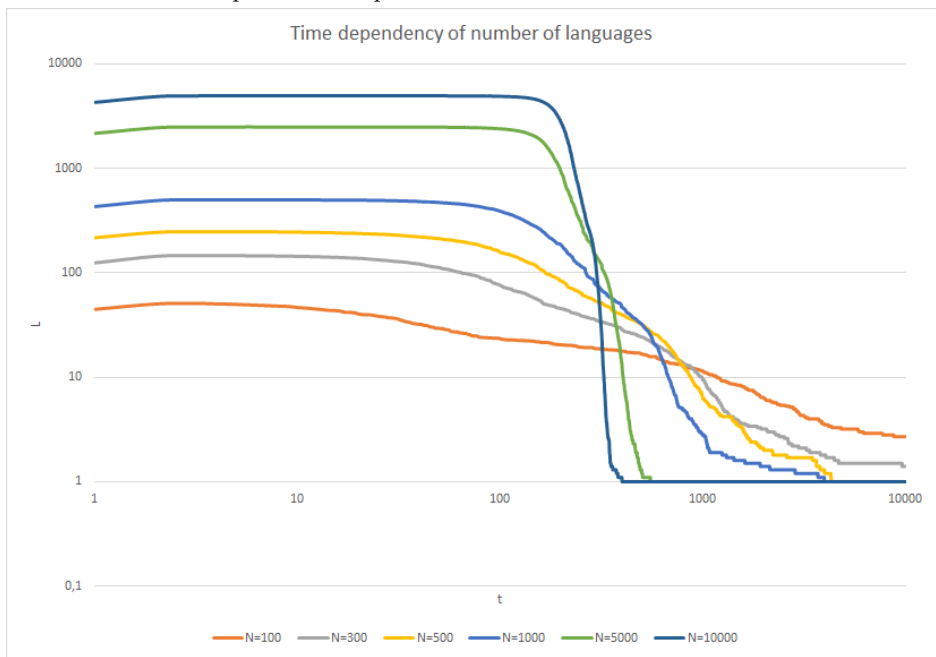
Figures 4.1 and 4.2 show how the results obtained from the experiments

¹<https://www.oracle.com/java/index.html>

4 Implementing the Naming Game Model on Adaptive Weighted Networks

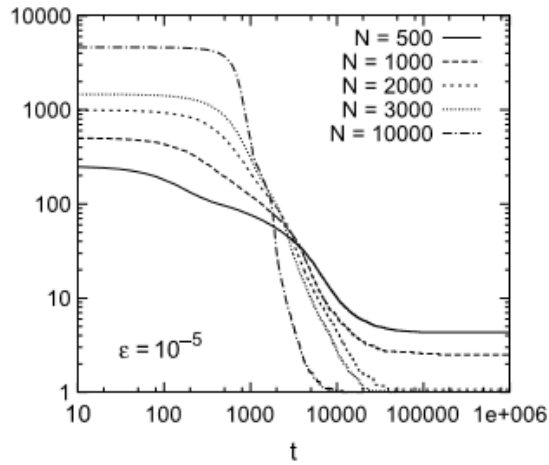


(a) Results from Lipowska and Lipowski (2012). Reprinted with permission.

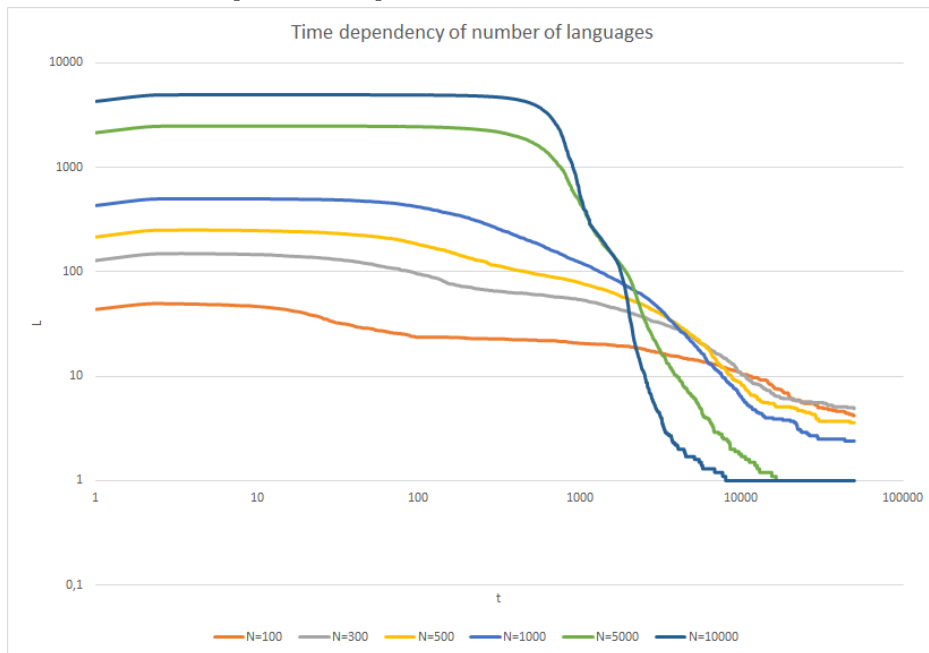


(b) Results from this thesis's experiments.

Figure 4.1: The time dependency of the number of languages L , calculated for $\varepsilon = 10^{-4}$ and for various numbers of agents N (logarithmic scale).



(a) Results from Lipowska and Lipowski (2012). Reprinted with permission.



(b) Results from this thesis's experiments.

Figure 4.2: The time dependency of the number of languages L , calculated for $\varepsilon = 10^{-5}$ and for various numbers of agents N (logarithmic scale).

in this thesis differ from Lipowska and Lipowski's results (note the different span in the number of generations). We see a very similar trend among the two sets of experiments.

Figure 4.3 shows how the success rate depends on the time (number of generations) when the product of the population size and the squared ε is kept constant. The general trend again seems very similar. The only noticeable difference is that the results are not identical in raw numbers: For example, for $N = 8000$, in Lipowska and Lipowski's research s drops to $0.55 - 0.6$ at its lowest (after the initial phase), while it in this thesis's research drops well under 0.50 . It cannot easily be established whether this is due to random fluctuations in the individual runs the graphs are based on or differences in the implementations.

Figure 4.4 shows how the number of languages depends on the time (number of generations) when the product of the population size and the squared ε is kept constant. Again, there are very few differences between the two graphs.

Figure 4.5 shows how the relative number of languages (the number of languages relative to the population size) depends on the time (number of generations) when the product of the population size and the squared ε is kept constant. The graphs are more or less identical.

Figure 4.6 compares how the share of the agents using the most common languages depends on the time (number of generations) when the product of the population size and the squared ε is kept constant. There are no notable differences to the results published by Lipowska and Lipowski.

A fact that has not been mentioned above, but is persistent across most of the plots, is that the implementations of this thesis show a steep increase in L for the first few generations. This can be explained easily: As the speaker in every dialog is drawn randomly, it will take a few generations until all agents have participated in at least one dialog and expanded their vocabulary to the size of 1 (initially all vocabularies are empty). As the graphs of Lipowska and Lipowski (2012) all begin with $t = 10$, this effect cannot be seen in their work, but there is no reason to believe it is not present.

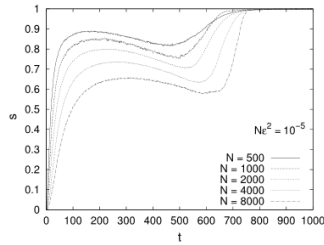
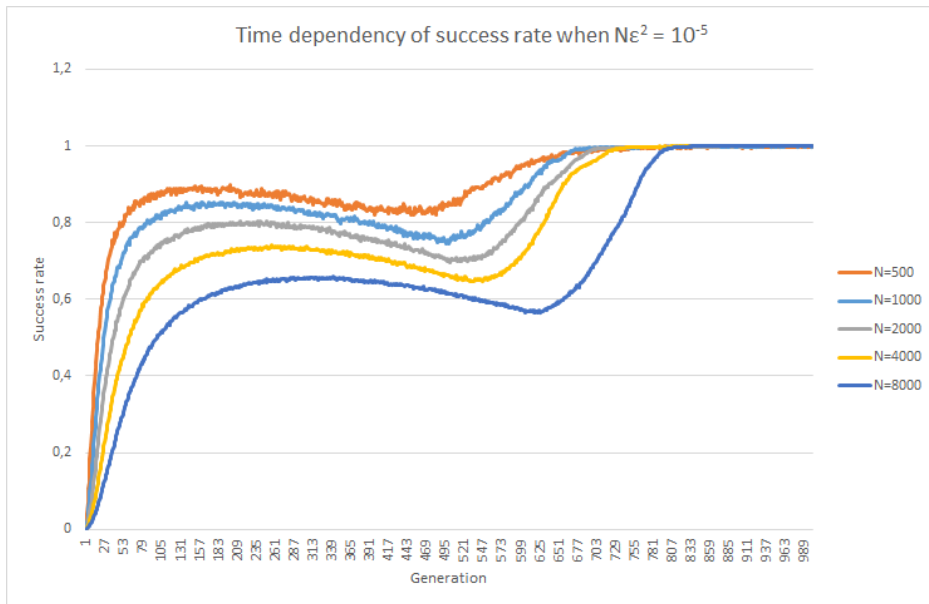


Figure 3. The time dependence of the success rate s calculated for several values of N and for $N\epsilon^2 = 10^{-5}$.

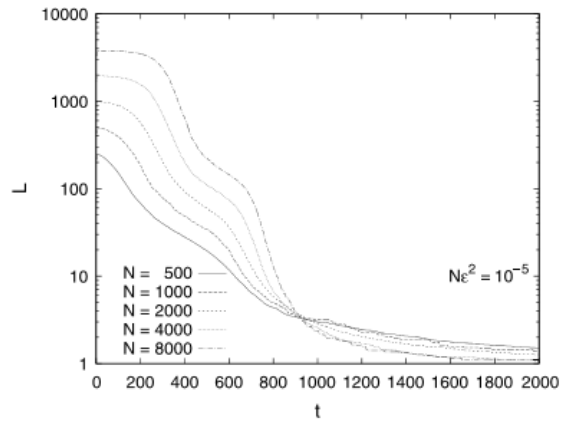
(a) Results from Lipowska and Lipowski (2012). Re-printed with permission.



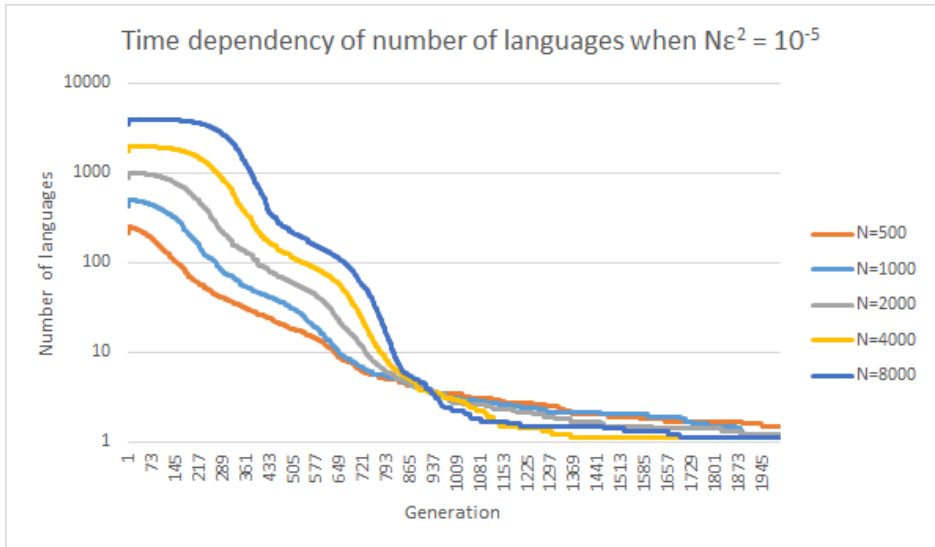
(b) Results from this thesis's experiments.

Figure 4.3: The time dependency of the success rate s , calculated for $N\epsilon^2 = 10^{-5}$ and for various numbers of agents N . **Left:** Results from Lipowska and Lipowski (2012)

4 Implementing the Naming Game Model on Adaptive Weighted Networks

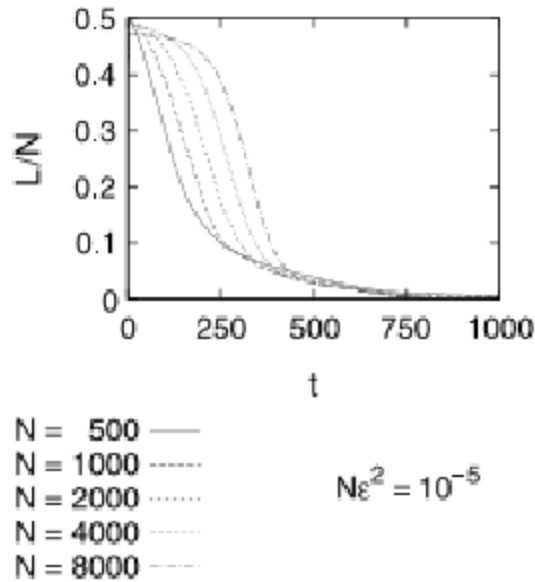


(a) Results from Lipowska and Lipowski (2012). Reprinted with permission.

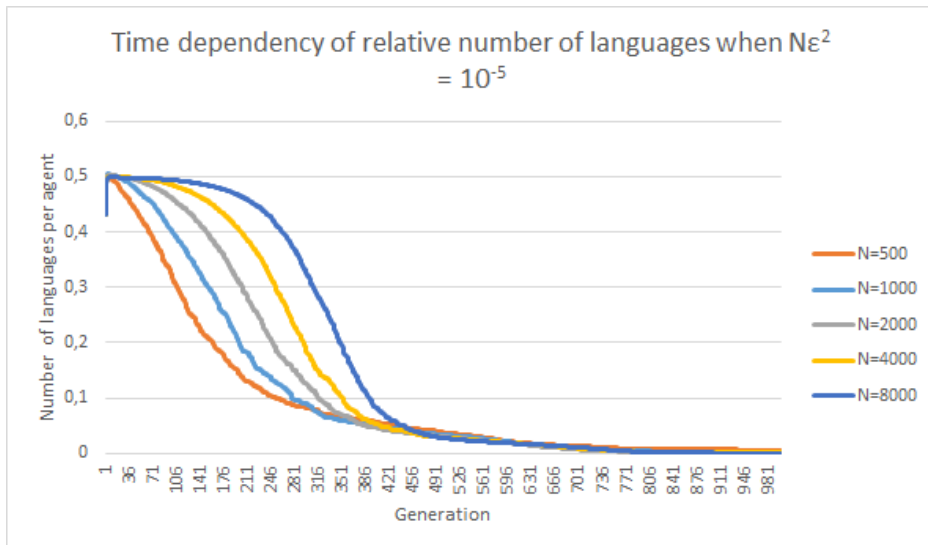


(b) Results from this thesis's experiments.

Figure 4.4: The time dependency of the number of languages L , calculated for $N\epsilon^2 = 10^{-5}$ and for various numbers of agents N (logarithmic scale).



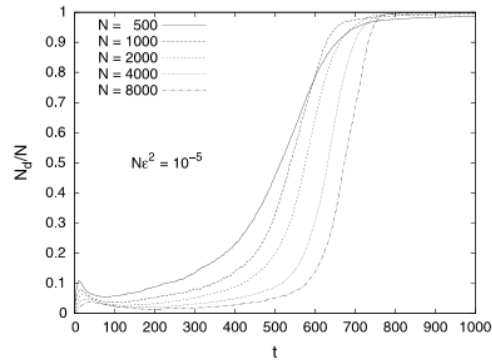
(a) Results from Lipowska and Lipowski (2012). Reprinted with permission.



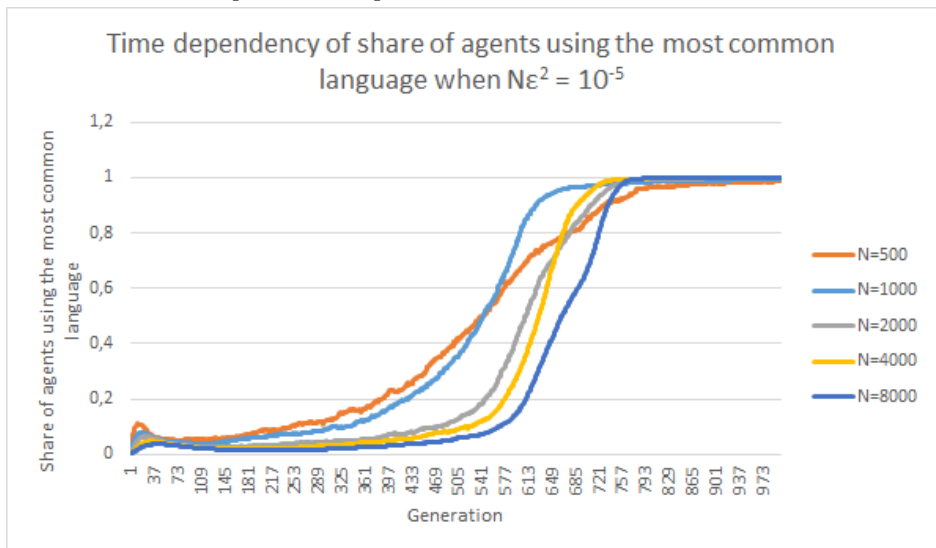
(b) Results from this thesis's experiments.

Figure 4.5: The time dependency of the number of languages relative to the population size, L/N , calculated for $N\epsilon^2 = 10^{-5}$ and for various numbers of agents N .

4 Implementing the Naming Game Model on Adaptive Weighted Networks



(a) Results from Lipowska and Lipowski (2012). Reprinted with permission.



(b) Results from this thesis's experiments.

Figure 4.6: The time dependency of the share of agents using the most common language, N_d/N , calculated for $N\epsilon^2 = 10^{-5}$ and for various numbers of agents N .

5 The Evolutionary Naming Game

In this chapter a new model is presented and examined. It has been given the name The Evolutionary Naming Game (ENG) for simplicity, but a more descriptive name could be The Naming Game on Adaptive Weighted Networks, in an Evolutionary, Multi-Conceptual Setting.

These experiments differ from the ones presented in Chapter 4 in the following manner:

- The pool of agents is renewed for each generation. This is done through reproduction: A successful communication results in the speaker and the hearer producing an offspring. When the number of generated children reaches the size of the agent pool, the children replace the existing agents and a new generation begins.
- An agent adopts the full vocabulary of its two parents, possibly with a small change (“mutation”).
- A newborn is given a social relation to other children and adult agents. This is done by the child “inheriting” the average number of dialogs and successful communications from its parents, between them and every other agent.
- The game is not restricted to words for one concept only. A few different setups regarding the number of existing concepts were tested. Each communication attempt involves one randomly chosen concept.

While Section 5.1 offers a much more detailed presentation of the model than the above bullet points, Section 5.2 presents the motivation for why the specifics of the model have been decided on. Section 5.3 describes the experimental setup. An simple example of a single experiment run is

5 The Evolutionary Naming Game

presented in Section 5.4. The initial experimental results are presented, mostly in tables and plots, in Section 5.5, along with a brief discussion. Some more experiment results are presented in Section 5.6, where the impact of the different parameters in simulation is examined. The main discussion of the results and how they compare to The Naming Game on Adaptive Weighted Networks (NGAWN) from Chapter 4 is placed in Chapter 6.

5.1 Architecture

This section will describe the details of the implementation. An outline can be found above.

NGAWN (Lipowska and Lipowski, 2012) serves as a basis for these experiments. That means that they are conducted within a population of agents trying to establish a common vocabulary, i.e., a traditional naming-game setting.

As opposed to NGAWN, there can be varying number of concepts in the environment. Several configurations will be tested.

1. N agents are generated. These agents have empty an vocabulary for all existing concepts.
2. Until N children are generated, a random speaker is chosen for a dialog.
3. A hearer is chosen for the dialog, based on Formula 5.1.
4. The speaker utters a randomly chosen word from his vocabulary for a randomly chosen concept.
5. If the speaker has no words for the concept, a new word is generated.
6. Whether the hearer has the word in its vocabulary determines if the dialog is a success or not.
7. If the dialog is a failure, the hearer adds the word to its vocabulary.
8. If the dialog is a success, both the speaker and the hearer retain only the spoken word in their vocabularies.

9. In case of a successful dialog, a child is also generated, with the speaker and the hearer as parents.
10. The child inherits all words for all concepts from its parents, but for each word there is a given probability of a changed word — the equivalent of a mutation.
11. The child also inherits the number of dialogs and successes for all other existing agents from its parents, further elaborated below.
12. When N children are generated, the children replace the adult agents and a new generation begins. The game runs for t generations.

$$P(b) = \frac{w(a, b) + \varepsilon}{\sum_x (w(a, x) + \varepsilon)} \quad (5.1)$$

Formula 5.1 shows the probability of speaker a choosing hearer b based on the weights w ($\frac{\text{number of dialogs}}{\text{number of successes}}$). (The formula is identical to Formula 4.1, but is repeated for the sake of convenience.)

A note on the inheritance of number of dialogs and successes (social weights): Every time a new agent is generated, its number of dialogs with another agent a is set to the average number of dialogs between the agent's two parents and a . The same happens for the number of successful dialogs. This processes is repeated for all existing agents, including children generated up until this point. As all weights are symmetric, i.e., $w(a, b) = w(b, a)$, this means that all children eventually will have weights between each other in place. The reason it has to be done for the old generation of agents as well, is for these agent to have a relation to the child in question in case they themselves become parents later in the generation, and thus need to pass on their weights for the child (and all other agents) to their own child.

A detailed breakdown of the procedures is offered as pseudo-code. Algorithm 5 is the starting point of the simulation of a dialog. The other algorithms presented are called by the main algorithm. Note that Algorithms 7, 8 and 9 are mostly identical to their counterparts for NGAWN, presented in Section 4.1.

Algorithm 5 ENG, general simulation

Input: t , the number of generations

Input: N , the number of agents

Input: c , the number of concepts

- 1: $agents \leftarrow$ a set of agents with empty vocabularies, of size N
 - 2: $concepts \leftarrow$ a set of concepts of size c
 - 3: $dialogs \leftarrow$ an array with the number of performed dialogs for each agent-pair
 - 4: $successes \leftarrow$ an array with the number of successful dialogs for each agent-pair
 - 5: **for** t generations **do**
 - 6: $children \leftarrow$ an initially empty set of agents
 - 7: **while** $children.size < N$ **do**
 - 8: $children \leftarrow$ **game**($agents$, $dialogs$, $successes$, $concepts$, $children$)
 - 9: **end while**
 - 10: $agents \leftarrow children$
 - 11: **end for**
-

5.2 Motivation and Justification

The changes from NGAWN to ENG are not arbitrary, and a justification of them is attempted in this section.

According to Bickerton (2008), one of the main factors holding research on language evolution back, is the large differences between human languages and existing computational models. ENG is not by any means intended to serve the role as a realistic framework for human language evolution, but some steps are taken to get closer to how language works in human societies. Many aspects are still unrealistic: For example it often take quite some more than one successful communication for two people to have child.

As opposed to NGAWN, ENG takes place in an evolutionary setting. The thought behind this is that in order to look at language evolution in a way relating to human languages' evolution, one has to do so in an evolutionary setting. One of the most defining features of ENG is the replacement of the entire agent pool for every generation. Evolution in

Algorithm 6 The `game` function of ENG

Input: *agents*, a set of agents with vocabularies**Input:** *dialogs*, an array with the number of performed dialogs for each agent-pair**Input:** *successes*, an array with the number of successful dialogs for each agent-pair**Input:** *concepts*, a set of concepts in the world**Input:** *children*, a set of children generated until now**Returns:** a boolean saying if the game was a success or not

```

1: concept  $\leftarrow$  randomly drawn concept from concepts
2: speaker  $\leftarrow$  randomly drawn agent from agents
3: hearer  $\leftarrow$  choose_hearer(speaker, agents, dialogs, successes)
4: word  $\leftarrow$  speak(speaker, concept)
5: success  $\leftarrow$  listen(hearer, word, concept)
6: if success then
7:   child  $\leftarrow$  produce_offspring(parents, concepts, agents, children)
8:   children.add(child)
9:   speaker.retain_only(word, concept)
10:  hearer.retain_only(word, concept)
11: else
12:  hearer.vocabulary[concept].add_word(word)
13: end if
14: dialogs[speaker][hearer]  $\leftarrow$  dialogs[speaker][hearer] + 1
15: if success then
16:  successes[speaker][hearer]  $\leftarrow$  successes[speaker][hearer] + 1
17: end if
18: return success

```

human languages is largely driven by the replacement of generations, as there exists evidence supporting most language learning and changes in an individual’s vocabulary being done at an early age, often referred to as “the critical period” or “the window of opportunity” (Pinker, 1994). The replacement is still a simplification, of course, as there at any time is more than one generation inhabiting the Earth.

The agents inherit (or “learn”) both their parents’ entire vocabulary. Other arrangements were considered, but this one stood out as the most

Algorithm 7 The `choose_hearer` function of ENG

Input: *speaker*, a chosen speaker for the dialog

Input: *agents*, a set of agents with vocabularies

Input: *dialogs*, an array with the number of performed dialogs for each agent-pair

Input: *successes*, an array with the number of successful dialogs for each agent-pair

Returns: a hearer for the dialog from *agents*

- 1: $weights \leftarrow$ sum of $successes[speaker][agent]/dialogs[speaker][agent]$ for all *agent* in *agents*
 - 2: $random \leftarrow$ a random real number between 0 and $weights$
 - 3: $temp \leftarrow 0$
 - 4: **for all** *agent* \neq *speaker* in *agents* **do**
 - 5: $success_ratio = successes[speaker][agent]/dialogs[speaker][agent]$
 - 6: $temp \leftarrow temp + success_ratio + \varepsilon$
 - 7: **if** $temp > random$ **then**
 - 8: **return** *agent*
 - 9: **end if**
 - 10: **end for**
-

Algorithm 8 The `speak` function of ENG

Input: *speaker*, an agent to utter a word

Input: *concept*, the concept to reference

Returns: a word to be used in the dialog

- 1: **if** $speaker.vocabulary[concept]$ is empty **then**
 - 2: $speaker.vocabulary[concept].add_word(new_word)$ #
 Randomly generated word
 - 3: **end if**
 - 4: **return** randomly chosen word from $speaker.vocabulary[concept]$
-

sensible. Human children who grow up with parents speaking different languages, indeed do learn both their parents' languages at a native level (Paradowski and Bator, 2016). The approach of this model is thus in line with reality.

The vocabularies are as mentioned above subject to mutation. Spe-

Algorithm 9 The **listen** function of ENG

Input: *hearer*, an agent to hear a word**Input:** *word*, an uttered word**Input:** *concept*, the concept referenced**Returns:** a boolean saying if the hearer understood the word or not

```

1: if hearer.vocabulary[concept].contains(word) then
2:   return true
3: else
4:   return false
5: end if

```

cifically, for each word that is inherited, there is a fixed probability that a single letter in the word will be altered, e.g., ADELE may become IDELE. Such changes do happen at the societal level in reality, and they must necessarily originate from the individual level. However, in the real world which “mutations” happen, or at least which ones get spread in the population, is a phenomenon governed by what is called sound laws (Anttila, 1989).

When it was initially decided that the agent pool would be replaced regularly, no attempt to transfer the social relations between generations was intended. However, this approach undermines the whole point of developing different social weights between the agents: If these weights do not get the time to develop properly, but are instead reset as soon as a number of dialogs are completed, communities are not given the chance to form, and one of the most interesting parts of NGAWN is removed. Thus, one needs to somehow maintain some social relations across a number of generations. Now how is this done in the real world? Our world consist of billions of people. There are indeed social relations governing the who communicates with whom — the people one person speaks to during a day heavily depends on that person’s acquaintances, i.e., his or her social relations. These are in turn not arbitrary. It can be argued that they ultimately often stem from the parents, who introduce their child to many of their acquaintances, as well as raising it into a specific community at a specific physical location.

A trivial solution of how to pass on social weights through generations could be to simply add the weights from the two parents for each agent

Algorithm 10 The `produce_offspring` function of ENG

Input: *parents*, a list of the two agents producing an offspring**Input:** *concepts*, a list of the concepts in the simulation**Input:** *agents*, a list of adult agents**Input:** *children*, a list of generated children**Returns:** *child*, the new agent generated

```

1: child ← a new agent
2: for all concept in concepts do
3:   for all word in (agents[1].vocabulary[concept] ∪
   agents[2].vocabulary[concept]) do
4:     r ← a random real number between 0 and 1
5:     if r <  $\rho$  then
6:       word ← word with one letter modified
7:     end if #  $\rho$  is the mutation rate
8:     child.vocabulary.add_word(concept, word)
9:   end for
10: end for
11: for all agent in (agents ∪ children) do
12:   dialogs[child][agent] ← average of dialogs[parents[1]][agent] and
   dialogs[parents[2]][agent]
13:   successes[child][agent] ← average of successes[parents[1]][agent]
   and successes[parents[2]][agent]
14: end for
15: return child

```

in the world. However, if an average weight (or more precisely, either a number of dialogs or a number of successes) is n , a child of the next generation could count on getting weights of about $2n$. Their children would in turn obtain weights of $4n$, on average. The pattern is, of course, average weights in generation i of n^i . Anyone just slightly familiar with computational complexity will notice that this is a number that in the course of a couple dozen generations is unmanageable. As the average weights thus need to be kept relatively stable, the approach explained in Section 5.1 was decided on. Empirical data show that the average size of the weights do not vary a lot as generations pass by, but a slight increase can be seen. This is natural, as any dialog, successful or not, will increase

the number of attempted dialogs, and, possibly, the number of successful ones.

The last change that is made from NGAWN is that the world has been extended from one consisting of a single object to one consisting of any number of objects (though a small number has still been used in the simulations). Loreto et al. (2011) write that the naming game can be reduced to one with one concept only without the loss of generality. The reason for this is that as long as the number of available words is large enough to make the chance homonyms (the same word for two different concepts) negligible, the hearer does not have to guess which object the speaker is referring to — if he has it in his vocabulary, the game is a success. The setup of ENG would not offer any increased complexity in this respect even if homonyms were likely, seeing that the agent only checks its vocabulary for the object in question.

When the simulations nevertheless are done in a multi-concept environment, this is because it is still interesting to see the effects it has on the model. They turn out to be substantial. It must, however, be noted that in terms of realism, an increase from one to for example five concepts is not very significant, keeping in mind that real languages consist of words for thousands of different objects and concepts.

5.3 Experimental Setup

The goal of the research can be summarized as follows:

- Test whether a more realistic model is suitable for research, and if so:
- Find whether the model results in one or several co-existing language communities.
- Test different combinations of parameters to discover their effects.
- Determine if any configuration of the model compares to aspects of human language evolution.

All results are plotted on the basis of averaged results from nine independent runs, to avoid issues with unlikely results from single runs.

5 The Evolutionary Naming Game

L	The number of languages in the population
N	The number of agents in the population
N_i	The number of agents using the i th most popular word
t	The number of generations simulated
ε	Parameter used when speaker selects a hearer ^a .
s	The communicative success rate ^b
c	The number of concepts
ρ	The mutation rate when an agent learns from its parents

^aA larger ε means more variance, i.e., higher chance of choosing hearers with which the speaker has an unsuccessful communicative history (low relational weight). For the formula, see Formula 5.1

^bThe share of successful dialogs.

Table 5.1: Explanation of the parameters in the ENG experiments.

Table 5.1 explains parameters and symbols used in this and following sections of the chapter.

The initial experiments of Section 5.5 were done on a total of 10 configurations, presented in Table 5.2. As can read from the table, these experiments focus on different values of N and c . r and ε are held constant, at 0.03 and $\sqrt{\frac{10^{-5}}{N}}$, respectively. Section 5.6 will focus on the impact of different values of these parameters.

An overview of the experiments that were conducted can be found in Table 5.3. Each references which figure a specific plot for a specific configuration (using the numbering from Table 5.2) can be found.

5.4 An Example

To further clarify what happens in the course of a simulation, an example will be presented. The data stems from one run with the following parameters:

$$N = 10, c = 2, \varepsilon = \sqrt{\frac{10^{-5}}{N}} = 10^{-3}, \rho = 0.03.$$

N is kept smaller than in the other experiments to make it easy to track the changes, as well as keeping the size of the example within the scope of what is practical to present in this format.

Configuration	N	c
#1	20	1
#2	100	1
#3	200	1
#4	500	1
#5	100	2
#6	200	2
#7	500	2
#8	100	5
#9	200	5
#10	500	5

Table 5.2: The configurations used in the ENG experiments.

Configuration	$L(t)$	$N_i(t)^a$	$s(t)$	\mathbf{SW}^b	\mathbf{NG}^c
#1	5.1	5.4	5.14	5.17	5.20
#2	5.1	5.5	5.14	5.17	5.21
#3	5.1	5.6	5.14	5.17	5.22
#4	5.1	5.7	5.14	5.17	5.23
#5	5.2	5.8	5.15	5.18	5.24
#6	5.2	5.9	5.15	5.18	5.25
#7	5.2	5.10	5.15	5.18	5.26
#8	5.3	5.11	5.16	5.19	5.27
#9	5.3	5.12	5.16	5.19	5.28
#10	5.3	5.13	5.16	5.19	5.29

^a $t = 1, 2, 3$.

^bPercentile diagrams of social weights. Further explained in Section 5.5.4.

^c(Social) network graphs.

Table 5.3: Figure reference for Section 5.5.

5 The Evolutionary Naming Game

#	Agents	Weight	C	Vocab A	Vocab B	Word	S
1	7, 3	$\frac{0}{0} = 0$	1	–	–	RIKYC	N
2	10, 7	$\frac{0}{0} = 0$	2	–	–	FOMOP	N
18	6, 5	$\frac{0}{1} = 0$	1	JISAH	JISAH	JISAH	Y
34	7, 2	$\frac{1}{3} = 0.33$	1	POTOT RIKYC JALIQ	POTOT JISAH	JALIQ	N
64	6, 5	$\frac{7}{11} = 0.64$	2	NEFUQ SAJAJ TUNYV	NEFUQ	NEFUQ	Y

Table 5.4: Data for a selection of dialogs in ENG, generation 1.

#	Agents	Weight	C	Vocab A	Vocab B	Word	S
1	6, 8	$\frac{1.44}{2.06} = 0.65$	1	POTOT RIKYC JISAH	RIKYC JISAH	POTOT	N
2	4, 9	$\frac{1.59}{2.47} = 0.65$	2	NEFUQ	NEFUQ	NEFUQ	Y
5	5, 8	$\frac{1.56}{2.50} = 0.63$	2	NEFUQ	NEFUQ	NEFUQ	Y
9	6, 2	$\frac{1.44}{2.22} = 0.65$	1	POTOT RIKYC JISAH	JISAH	RIKYQ	N
12	3, 4	$\frac{1.59}{2.47} = 0.65$	2	NEFUQ	NEFUQ	NEFUQ	Y

Table 5.5: Data for a selection of dialogs in ENG, generation 3.

Tables 5.4, 5.5, 5.6, and 5.7 show a selection of the performed dialogs in the generations 1, 3, 20, and 100, respectively. *Agents* is the two agents in question — speaker and hearer, respectively. *Weight* is the number of successes divided by the number of dialogs, i.e., the social weight. *C* is the chosen concept. *Vocab A* and *Vocab B* are the vocabularies for the chosen concept for the speaker and the hearer, respectively. *Word* is the word uttered. *S* denotes whether the dialog was successful (Y) or not (N).

Table 5.8 shows key data for a selection of the generations. *Dialogs* is the total number of dialogs in the generation. *Successes* is the number of those that were successful. This number corresponds to *N*, because a

#	Agents	Weight	C	Vocab A	Vocab B	Word	S
1	8, 9	$\frac{1.31}{1.43} = 0.92$	1	JISAH	JOSAH JISAH	JISAH	Y
2	8, 3	$\frac{1.48}{1.62} = 0.92$	2	NEFUQ	NEFUQ	NEFUQ	Y
4	4, 6	$\frac{1.80}{1.95} = 0.92$	1	JISAH	JISAH	JISAH	Y
7	7, 3	$\frac{1.60}{1.75} = 0.91$	2	NEFUQ	NEFUQ	NEFUQ	Y
10	9, 1	$\frac{2.33}{2.46} = 0.95$	2	NEFUQ	NEFUQ NEFUG	NEFUQ	Y

Table 5.6: Data for a selection of dialogs in ENG, generation 20.

#	Agents	Weight	C	Vocab A	Vocab B	Word	S
1	1, 7	$\frac{1.66}{1.69} = 0.99$	1	JISAH	JISAH	JISAH	Y
2	10, 7	$\frac{1.54}{1.58} = 0.98$	2	NEFUQ	NEFUQ NOFUQ NEXUQ NENUQ NEXUF FEFUQ	NEFUQ	Y
6	9, 6	$\frac{2.11}{2.13} = 0.99$	2	NEFUQ NOFUQ JEFUQ NEXUQ NENUQ NEXUF FEFUQ	NEFUQ NOFUQ NEXUQ NENUQ NEXUF FEFUQ	FEFUQ	Y
9	3, 6	$\frac{1.72}{1.74} = 0.99$	2	NEFUQ NOFUQ NEXUQ NENUQ NEXUF FEFUQ	FEFUQ	NOFUQ	N
11	7, 2	$\frac{2.66}{2.69} = 0.99$	1	JISAH	JISAH	JISAH	Y

Table 5.7: Data for a selection of dialogs in ENG, generation 100.

5 The Evolutionary Naming Game

#	Dialogs	Successes	Words 1	Words 2
1	64	10	6	9
2	14	10	3	2
3	12	10	5	2
4	12	10	5	1
20	10	10	1	2
50	13	10	3	3
100	11	10	1	7

Table 5.8: Data for a selection of generations in ENG.

generation is finished when N children are generation, and a new child is generated for each successful dialog. *Words 1* and *Words 2* is the total number of words existing for concepts 1 and 2, respectively.

This small example will not be analyzed in any detail, but a few comments can be made:

The number of dialogs and successes between the agent-pairs does not hold any intuitive meaning after the first generation is completed. The reason for this, of course, is that when a new child is generated, it inherits the average number of dialogs and successes for each other agent from its parents. For a justification of this process, see Section 5.2.

The effects of mutation can be seen clearly. The word NEFUQ seems to have been the subject of particularly many mutations, with some agents having as many as seven variations of it in their vocabularies.

All aspects of the example above are not necessarily representative for simulations with other parameters, e.g., a higher number of agents. For example it can be seen that in this example the success rate quickly approaches 1, surpassing 0.7 already in the second generation.

5.5 Initial Results

This section presents the results from the experiments mentioned in Table 5.2.

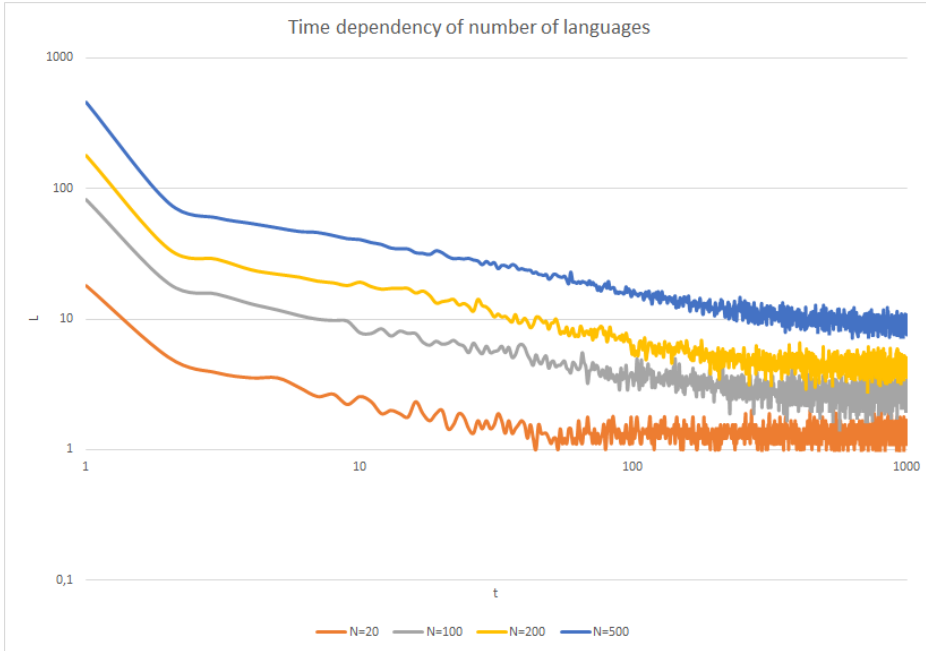


Figure 5.1: $L(t)$ in ENG. $c = 1$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$. Logarithmic scale.

5.5.1 Number of Languages in Population

The first set of results that is examined is the number of languages L in the population as a whole, as a function of the time (number of generations), t . More specifically, all words that appear in at least one agent’s vocabulary are counted. In the scenarios with more than one concept, the number of words is only counted across a single concept. While counting all combinations of words across all concepts would be interesting, and perhaps a more realistic measurement of the number of “languages”, it is highly impractical — with the variations offered by the configurations used in these experiments the number of permutations would be enormous.

Figures 5.1, 5.2 and 5.3 compare $L(t)$ for four different choices of N .

For a small number of concepts and agents, L relatively quickly converges towards 1. Because of random mutations between the generations it, however, never stabilizes at a single language.

5 The Evolutionary Naming Game

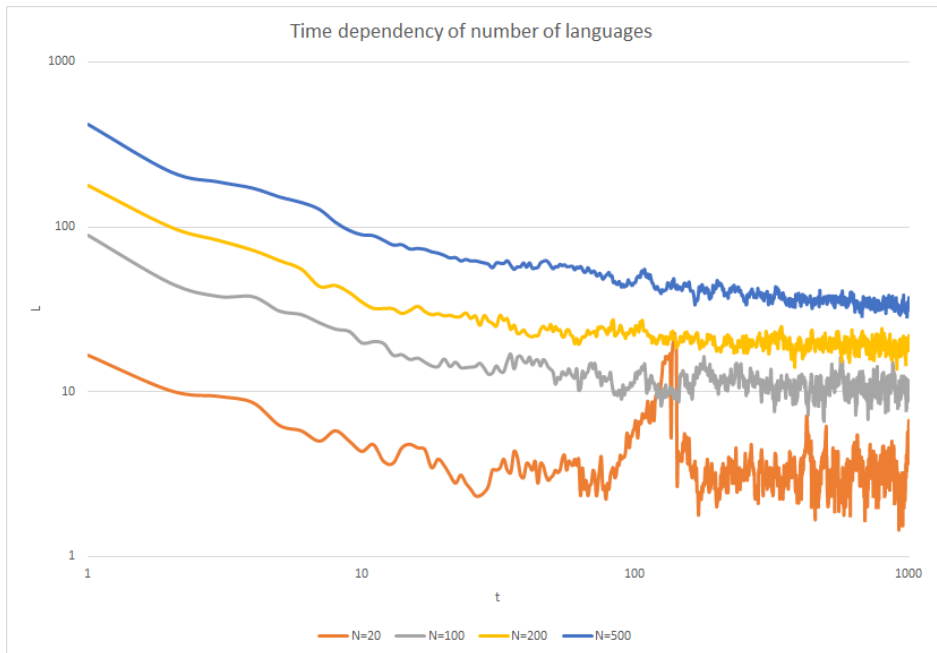


Figure 5.2: $L(t)$ in ENG. $c = 2$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$. Logarithmic scale.

5.5 Initial Results

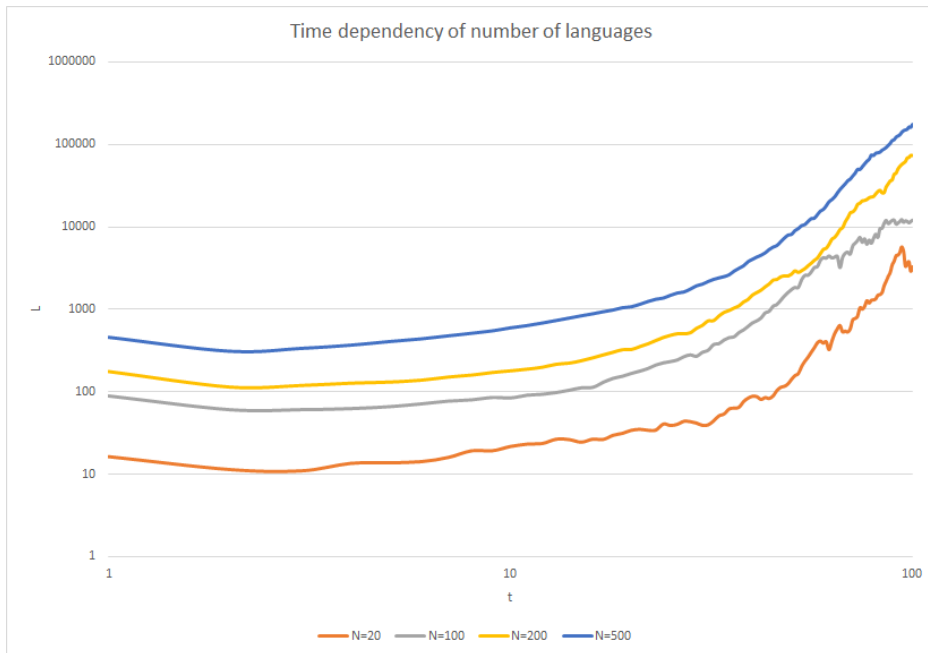


Figure 5.3: $L(t)$ in ENG. $c = 5$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$. Logarithmic scale.

5 The Evolutionary Naming Game

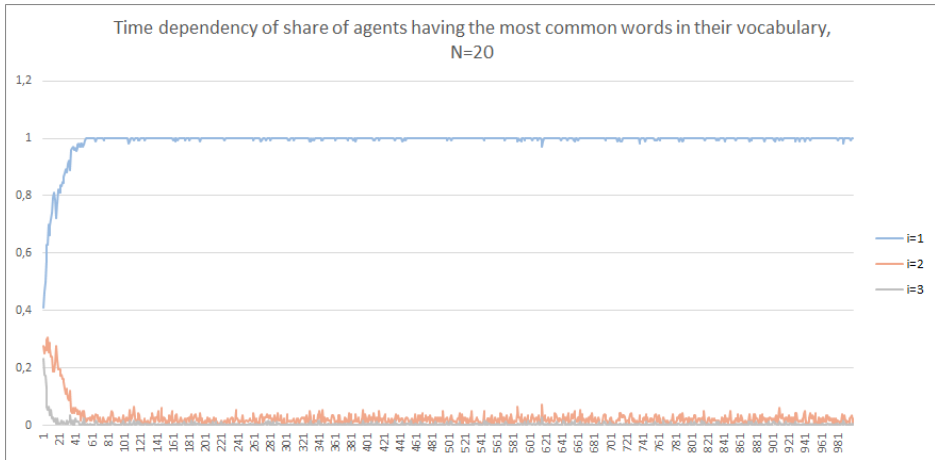


Figure 5.4: $N_i(t)$ in ENG. $N = 20$, $c = 1$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

For a large number of concepts (five), L grows and quickly reaches very high values. E.g., with $c = 5$ and $N = 500$, there are more than 100,000 unique words in the simulation after 100 generations. This means that each agent on average knows 2000 unique synonyms for a single concept! Simulating this scenario for 1000 generations is simply not computationally feasible, as the vocabularies continue to grow.

5.5.2 Popularity of Most Used Languages

This section features plots showing the popularity N_i of the three most used languages ($i = 1, 2, 3$), as a function of the number of generations t . This means that the frequencies of all words in the population are found, and the number of agents having each of those in their vocabulary is counted. Only occurrences across a single concept are plotted.

Figures 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12 and 5.13 each show $N_i(t)$ for a specific choice of N and c .

When the world consists of one and two concepts, eventually nearly 100 % of the agents have the most common word in their vocabularies. This trend is clear for all N , but the greater the number of agents, the more generations it takes.

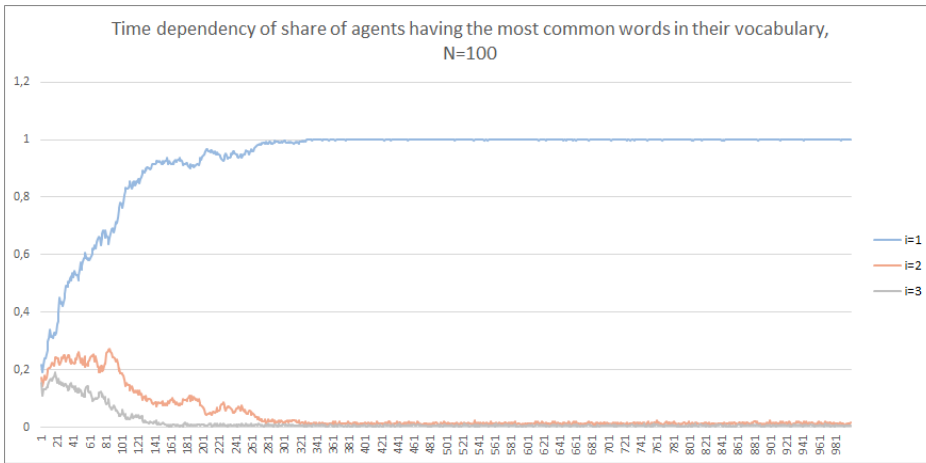


Figure 5.5: $N_i(t)$ in ENG. $N = 100$, $c = 1$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

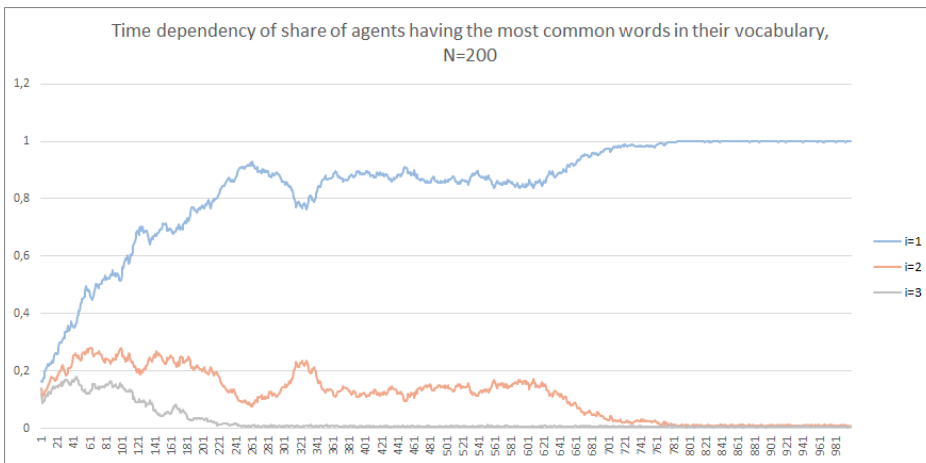


Figure 5.6: $N_i(t)$ in ENG. $N = 200$, $c = 1$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

5 The Evolutionary Naming Game

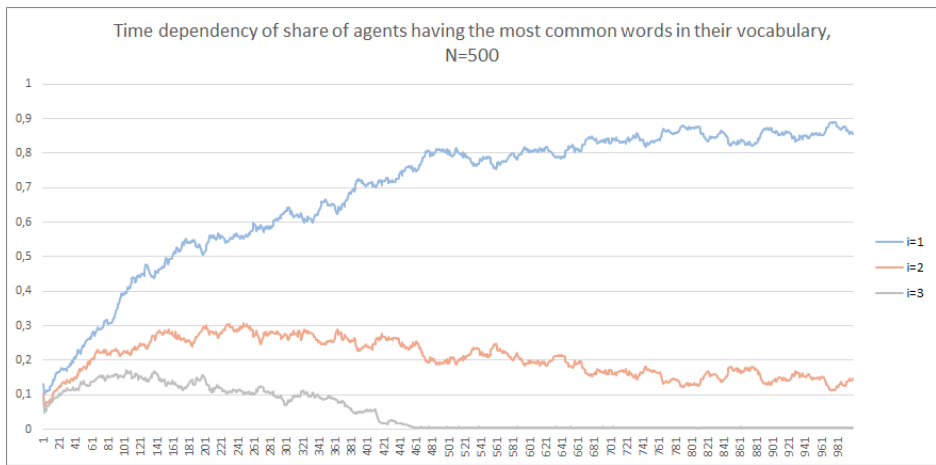


Figure 5.7: $N_i(t)$ in ENG. $N = 500$, $c = 1$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

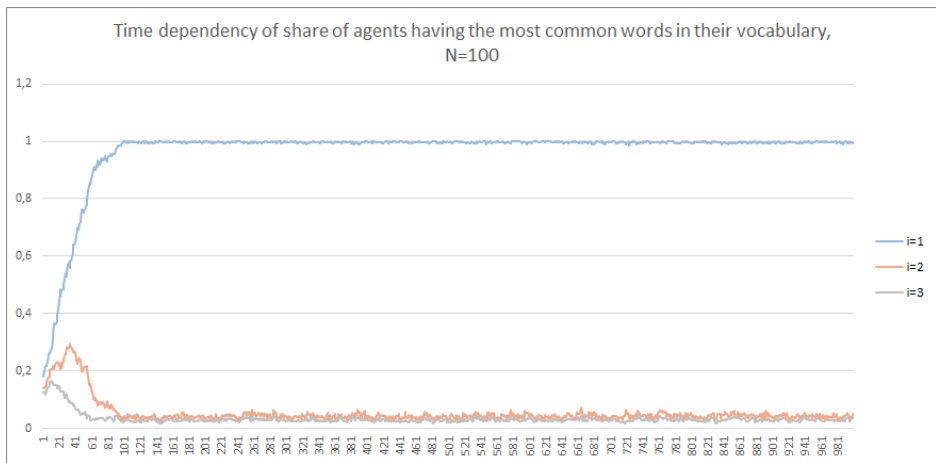


Figure 5.8: $N_i(t)$ in ENG. $N = 100$, $c = 2$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

5.5 Initial Results

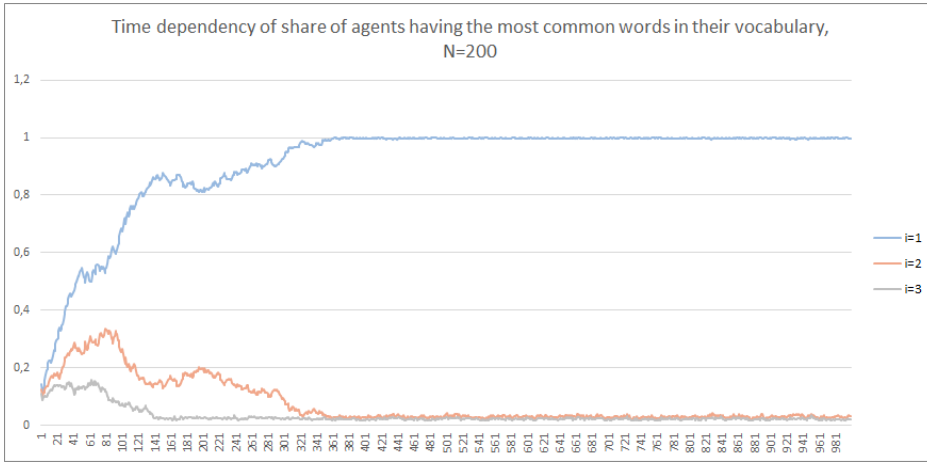


Figure 5.9: $N_i(t)$ in ENG. $N = 200$, $c = 2$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

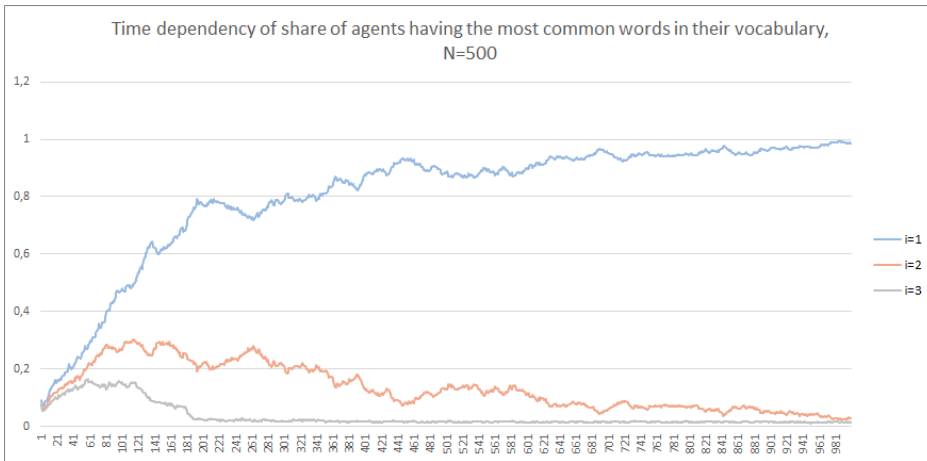


Figure 5.10: $N_i(t)$ in ENG. $N = 500$, $c = 2$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

5 The Evolutionary Naming Game

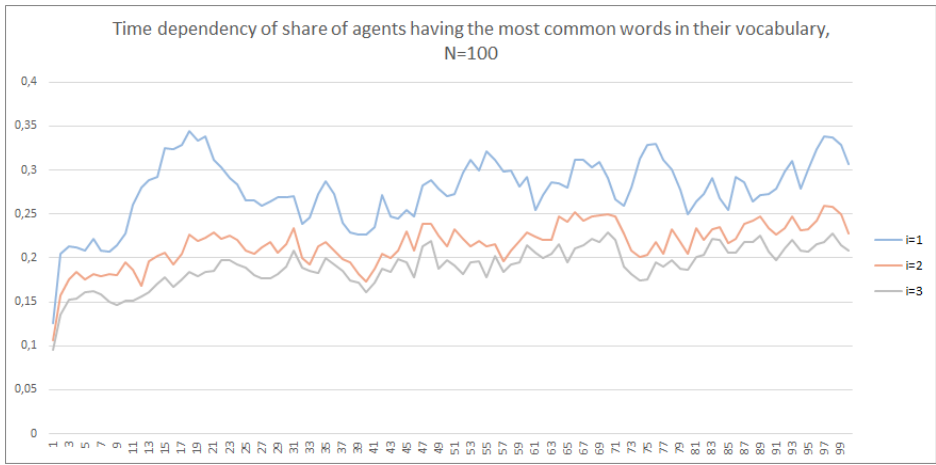


Figure 5.11: $N_i(t)$ in ENG. $N = 100$, $c = 5$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

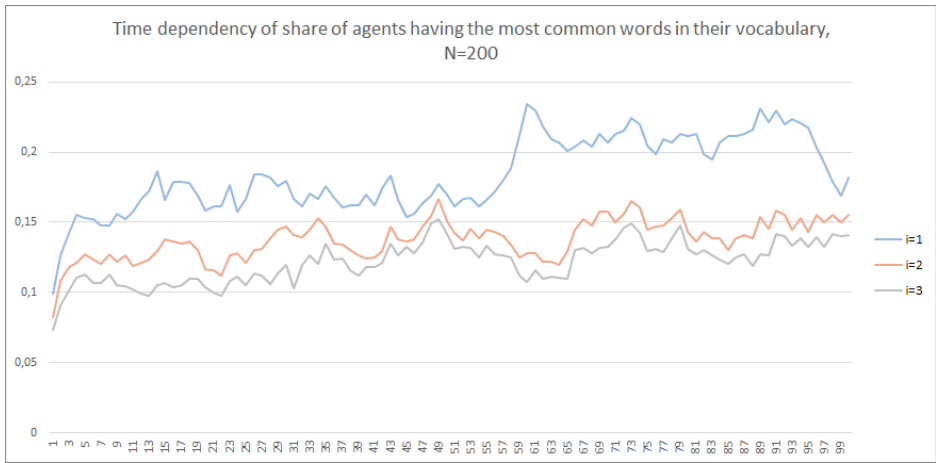


Figure 5.12: $N_i(t)$ in ENG. $N = 200$, $c = 5$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

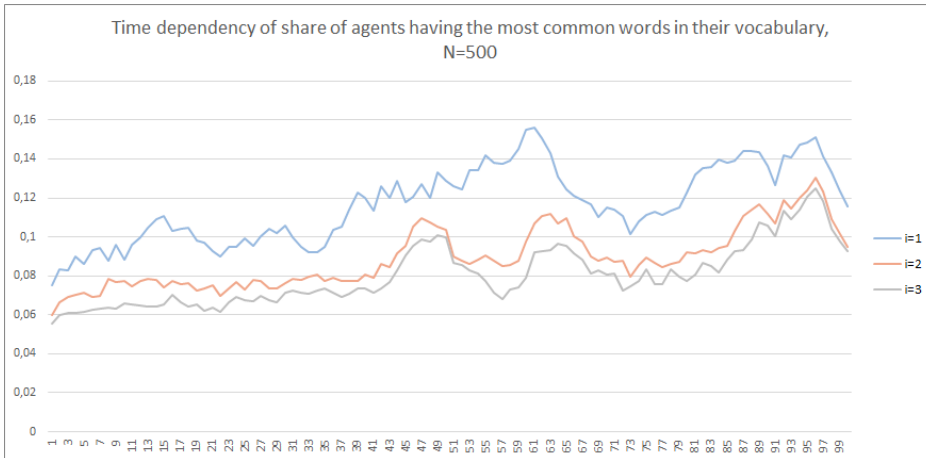


Figure 5.13: $N_i(t)$ in ENG. $N = 500$, $c = 5$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $i = 1, 2, 3$.

When $c = 5$, no word successfully makes its way into the vocabulary of more than at most 30 % of the agents. There is no considerable emergence of one word gaining popularity on behalf of others in these plots.

5.5.3 Success Rate

In this section the success rate s as a function of the number of generations t is examined. The definition of the success rate is simply the share of successful dialogs among all dialogs.

The results are depicted in Figures 5.14, 5.15 and 5.16. Each of them compare $s(t)$ for a varying number of agents N .

As can be seen in the figures, in every configuration involving 1 or 2 concepts s almost immediately reaches 1, meaning that practically all dialogs are successful. The smaller the N , the more variance there seems to be. This is most likely due to the fact that a single mutated word, which will almost always lead to a failed dialog (but could obviously lead to successful ones later on, when other agents have learned the mutated version of the word), having a greater relative impact when the population and total number of dialogs is smaller.

The last figure shows the situation when there are 5 concepts in the

5 The Evolutionary Naming Game

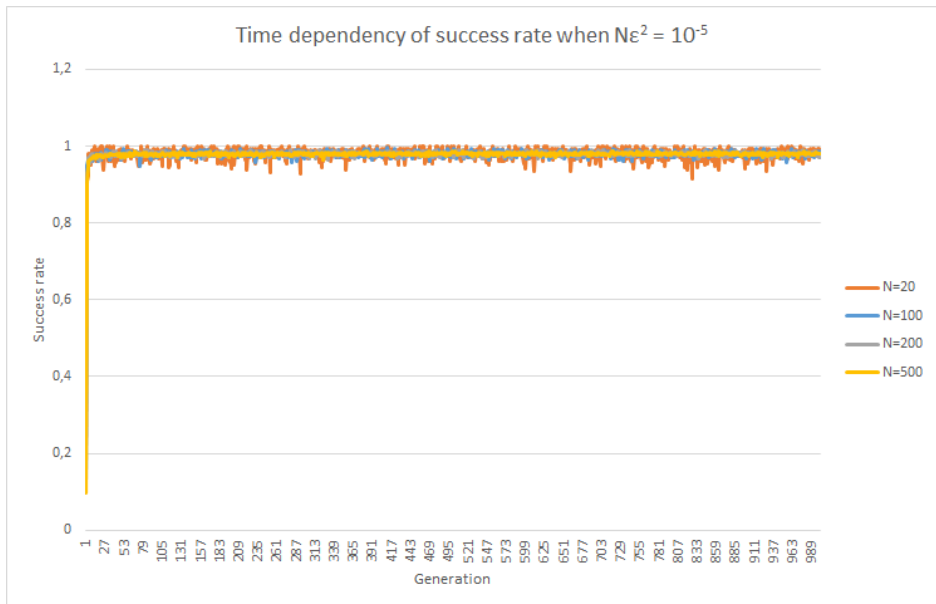


Figure 5.14: $s(t)$ in ENG. $c = 1$, $\epsilon = \sqrt{\frac{10^{-5}}{N}}$.

5.5 Initial Results

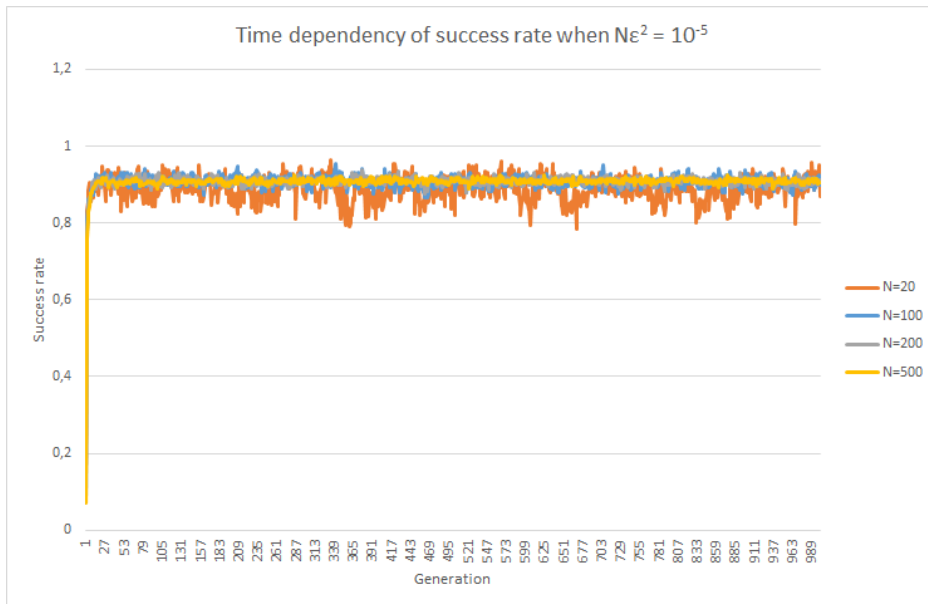


Figure 5.15: $s(t)$ in ENG. $c = 2$, $\epsilon = \sqrt{\frac{10^{-5}}{N}}$.

5 The Evolutionary Naming Game

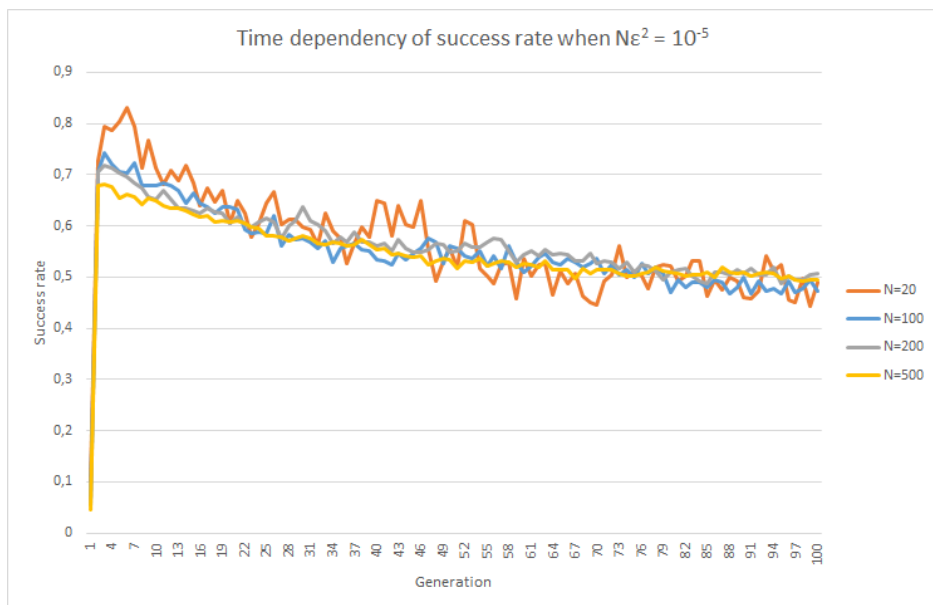


Figure 5.16: $s(t)$ in ENG. $c = 5$, $\epsilon = \sqrt{\frac{10^{-5}}{N}}$.



Figure 5.17: Percentile diagram showing social weights in ENG. $c = 1$, $t = 1000$.

world. The population size has no considerable impact here, as all configurations initially reach a success rate of ~ 0.7 , which quickly starts dropping, reaching a level where less than half of the dialogs are successful.

5.5.4 Social Weights

The figures in this section show what has been named percentile diagrams for the social weights. The “social weights” are the relationships between the number of successful dialogs and the number of total dialogs between two agents. In other words, they relate to the likelihood of choosing one particular hearer based on the speaker in a dialog. A social weight of 1 means that all dialogs between the agents in question have been successful, while a weight of 0 means that none of them have.

The diagrams in figures 5.17, 5.18 and 5.19 depict all social weights in the simulation divided into percentile, comparing different N s. Specifically:

1. $p_i =$ number of pairs j, k ($j \neq k$), such that $i - 1 < \frac{s_{j,k}}{d_{j,k}} < i$, where

5 The Evolutionary Naming Game

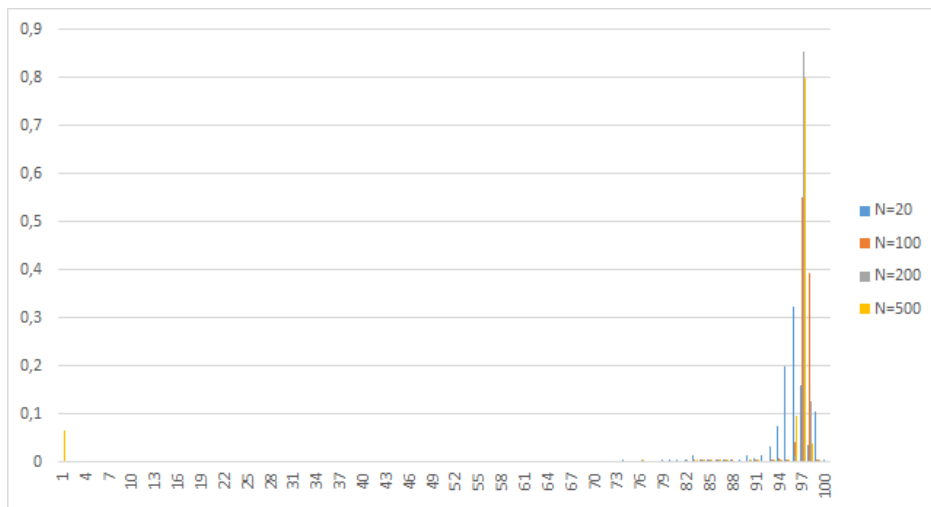


Figure 5.18: Percentile diagram showing social weights in ENG. $c = 2$, $t = 1000$.

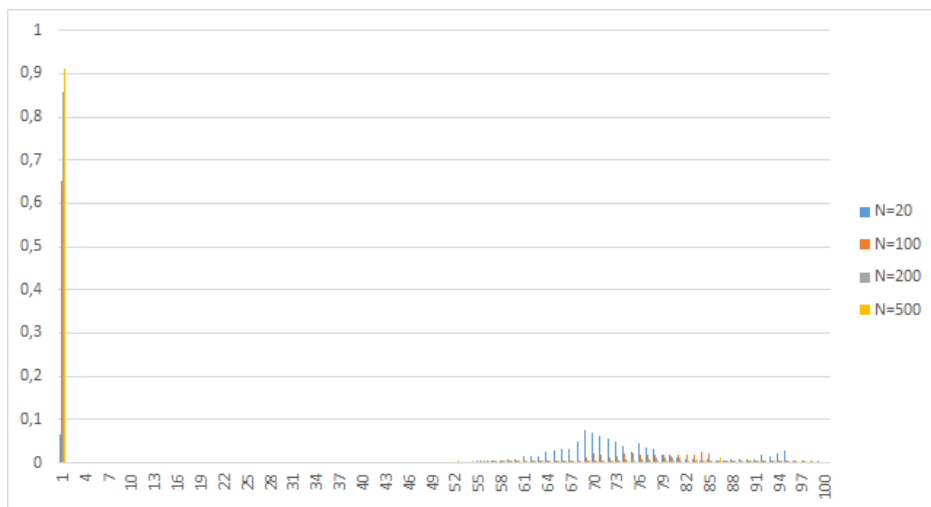


Figure 5.19: Percentile diagram showing social weights in ENG. $c = 5$, $t = 100$.

- p_i is the size of percentile i .
2. j and k are agents.
 3. $s_{j,k}$ and $d_{j,k}$ are the number of successful dialogs and total dialogs among the agent-pair (j, k) , respectively.

When the number of concepts is low (≤ 2), the number of weights below 0.9 is negligible. This points towards a high number of successful dialogs, as already shown above.

Again the story is different when the number of concepts is increased to 5. The majority of the weights are in the first percentile, i.e., the interval $[0, 1)$. However, some substantially higher weights can be seen.

5.5.5 Social Network Graphs

The last set of results is the social network graphs. In these graphs, all agents are shown as dots (i.e., vertices), and all social weights $w > 0.5$ are shown as lines between agents (i.e., edges). One cluster — a set of agents connected to each other — can be interpreted as a community with high internal intelligibility.

As opposed to the plots and graphs in the sections above, the ones presented here are not averaged over several runs, but rather snapshots from one single run. This could doubtlessly lead to certain graphs being less representative of the typical case than what would be optimal. However, there is no obvious way to present averaged versions of these network graphs.

Figures 5.20, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28 and 5.29 each consist of up to four separate images. They show how the social networks evolve over time. The images 1–4 (numbered in the order *upper-left*, *upper-right*, *lower-left*, *lower-right*) show the situation after 10, 50, 100 and 200 generations, respectively. One figure covers one specific configuration of N and c .

Clusters are formed in all cases. After a certain amount of time, all agents are members of a cluster for all the configurations. In several of the configurations tested all agents form one single cluster within the 200 generations depicted. The speed of this transition seems to be connected

5 The Evolutionary Naming Game

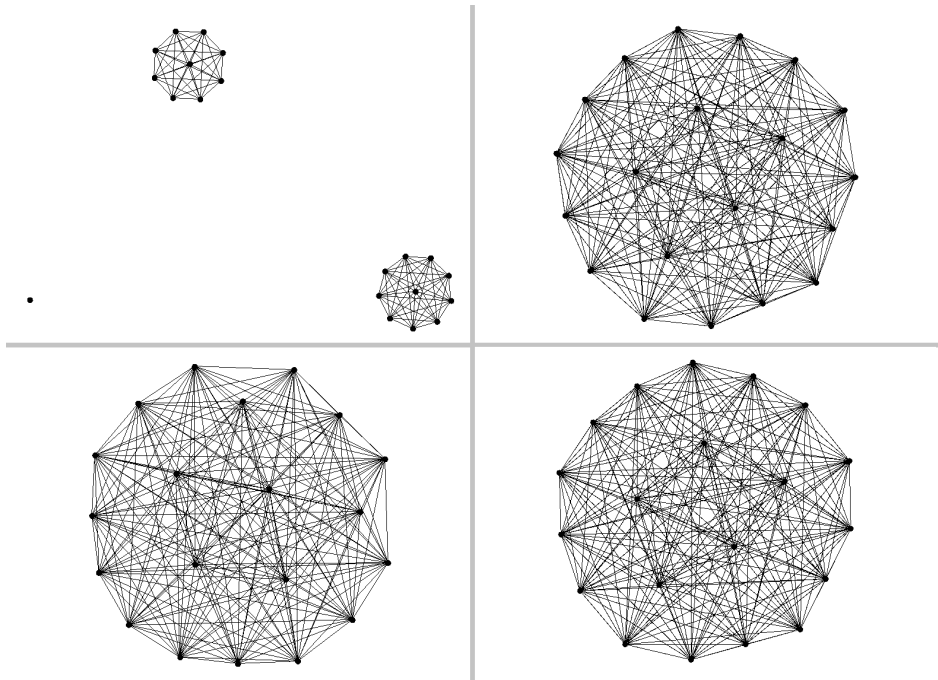


Figure 5.20: Social network graph for ENG. $N = 20$, $c = 1$. **1:** $t = 10$. **2:** $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

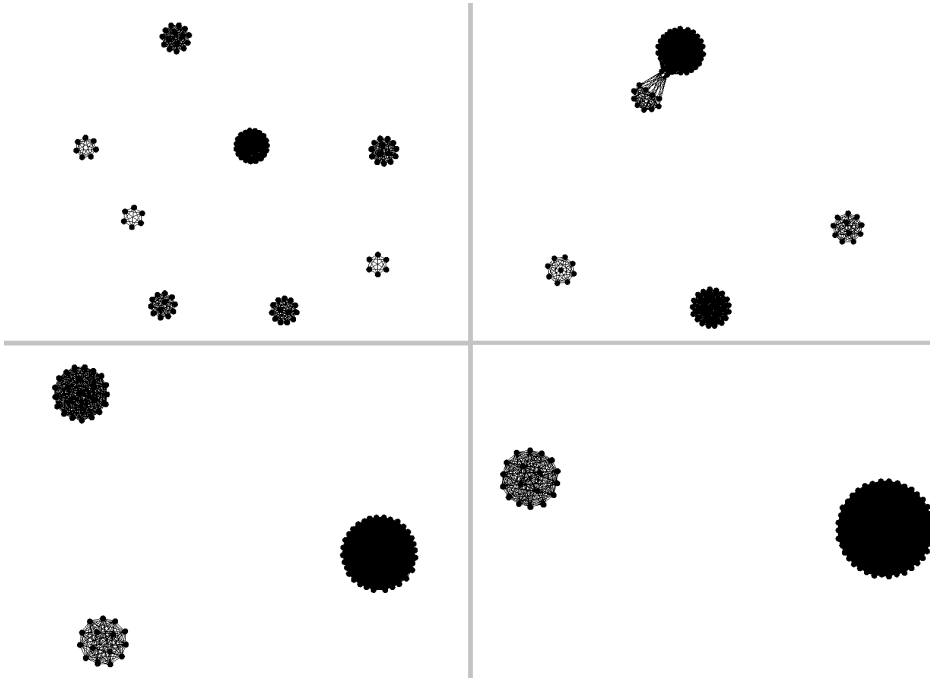


Figure 5.21: Social network graph for ENG. $N = 100$, $c = 1$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

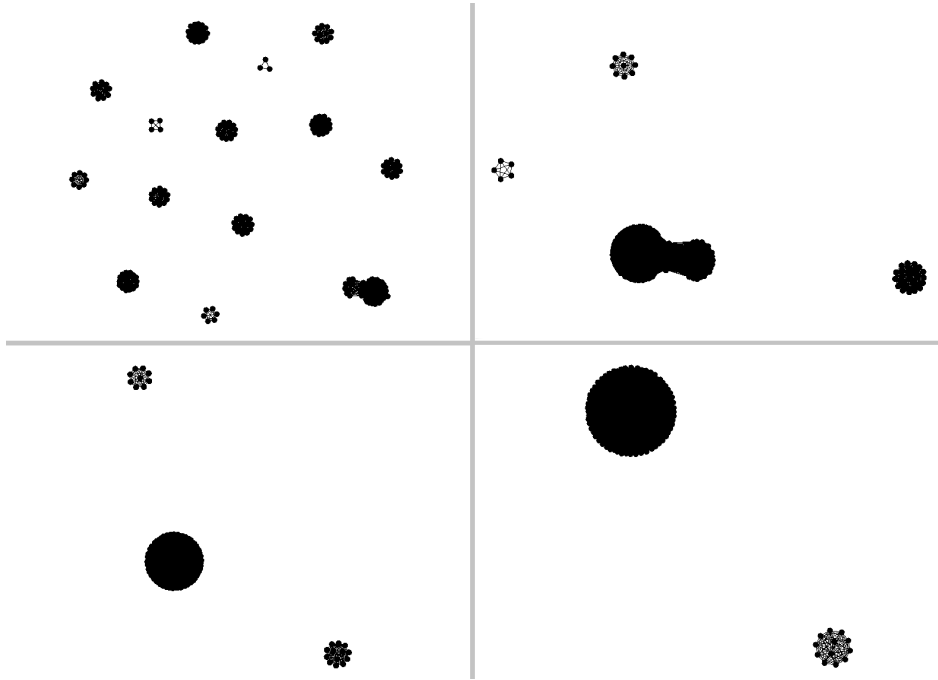


Figure 5.22: Social network graph for ENG. $N = 200$, $c = 1$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

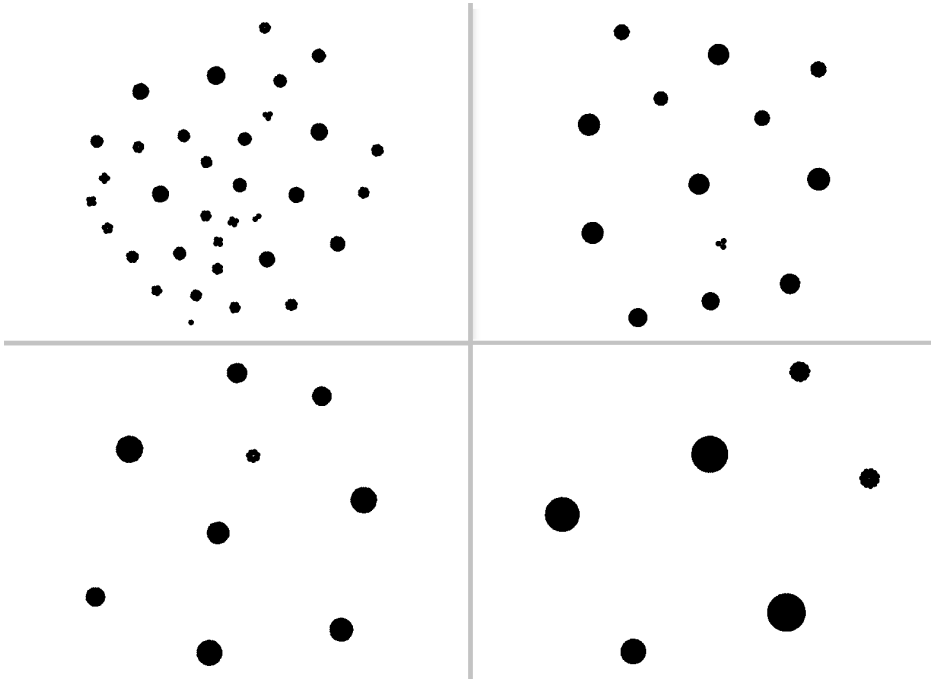


Figure 5.23: Social network graph for ENG. $N = 500$, $c = 1$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

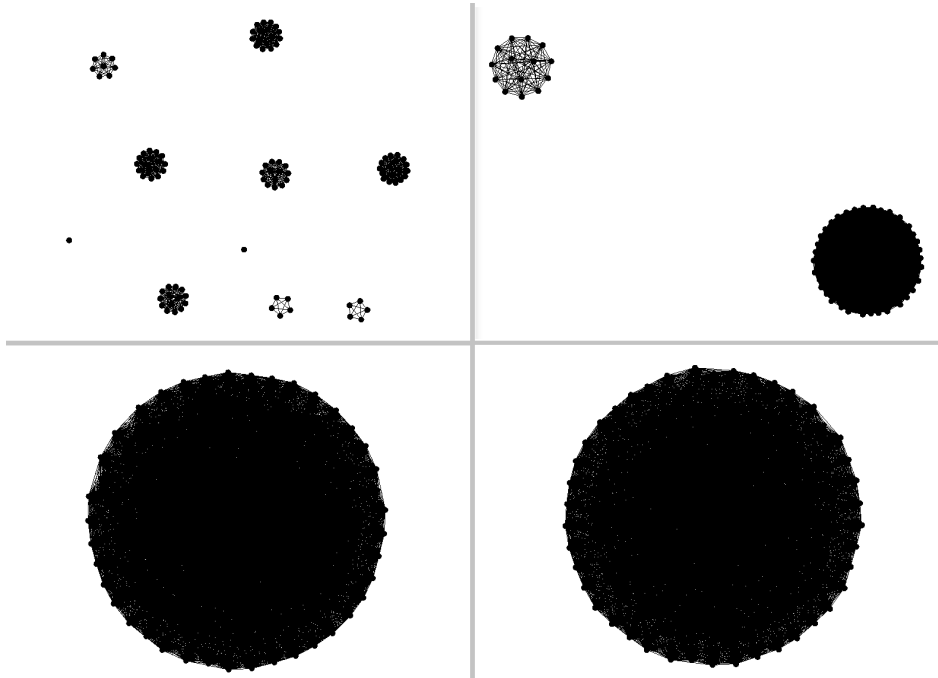


Figure 5.24: Social network graph for ENG. $N = 100$, $c = 2$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

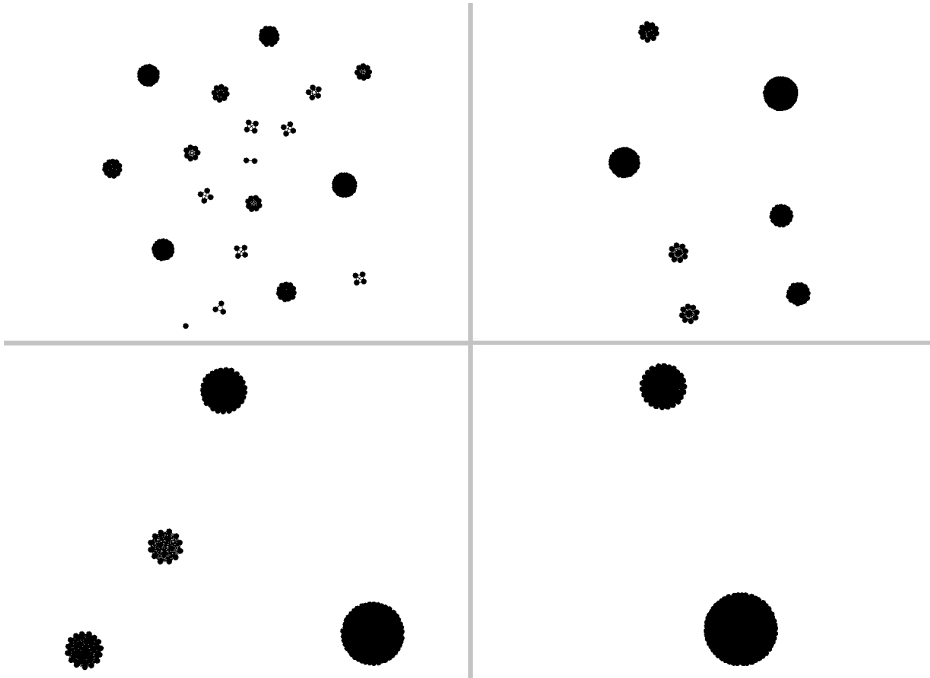


Figure 5.25: Social network graph for ENG. $N = 200$, $c = 2$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

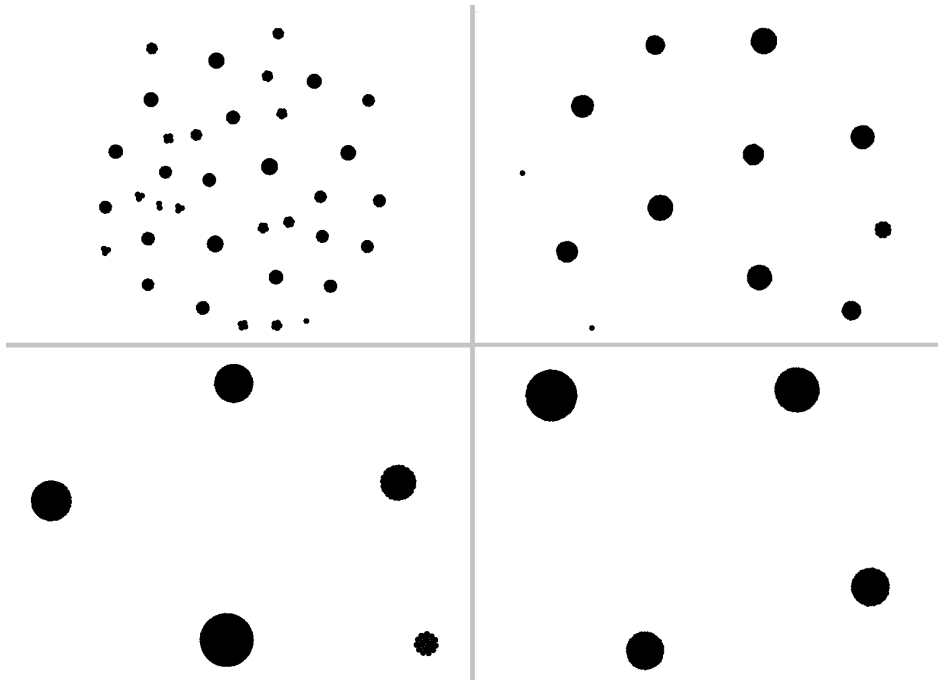


Figure 5.26: Social network graph for ENG. $N = 500$, $c = 2$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

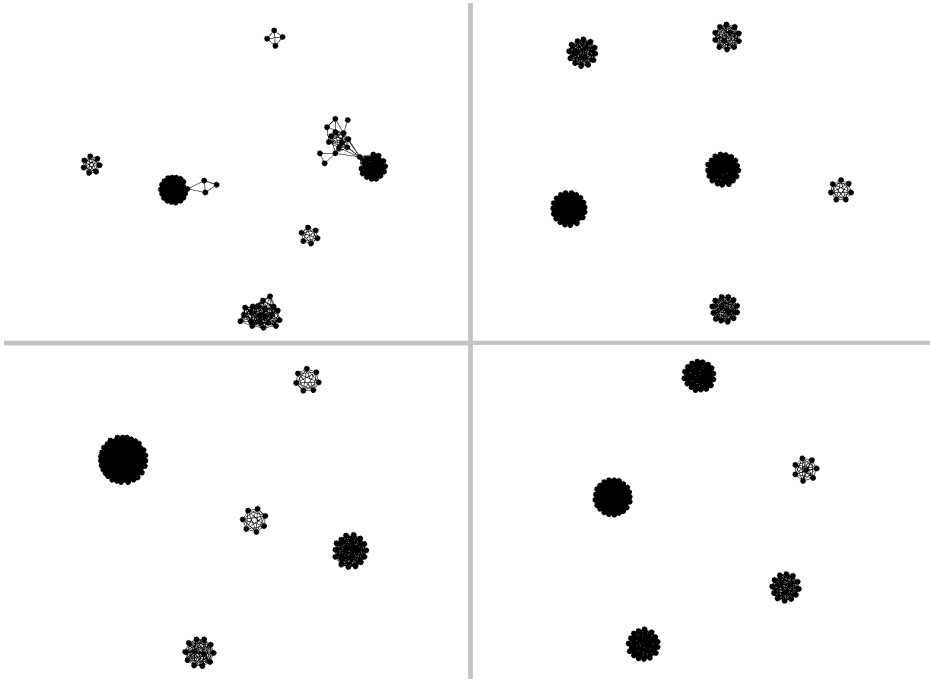


Figure 5.27: Social network graph for ENG. $N = 100$, $c = 5$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

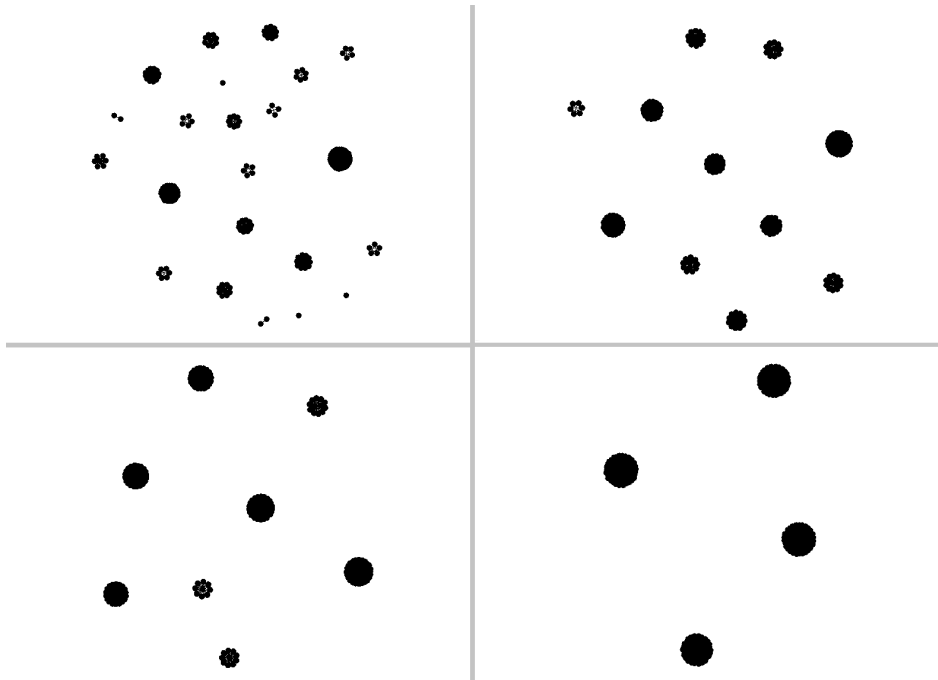


Figure 5.28: Social network graph for ENG. $N = 200$, $c = 5$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$. **4:** $t = 200$.

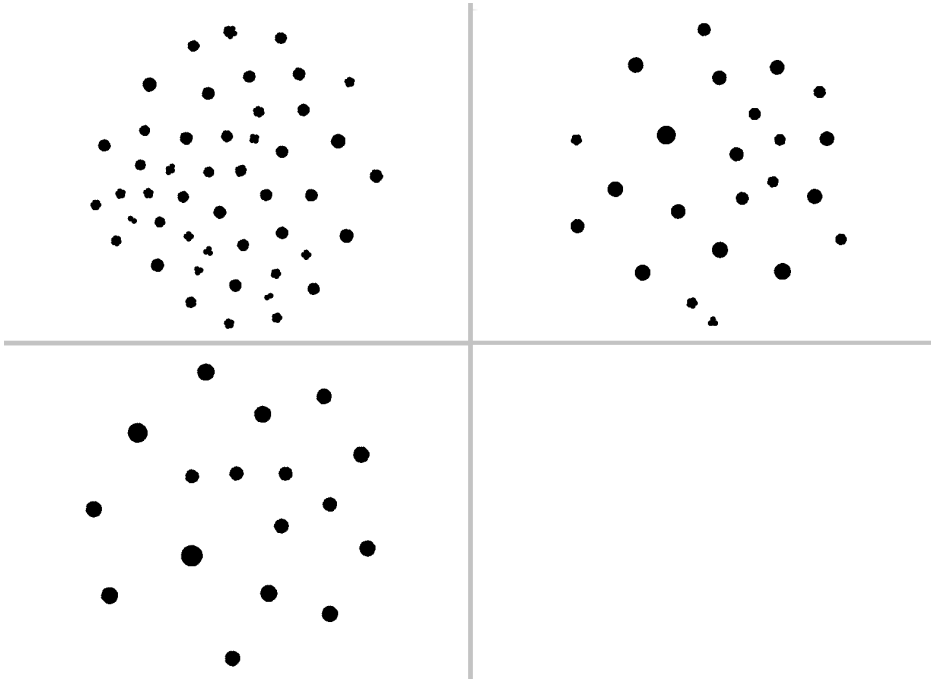


Figure 5.29: Social network graph for ENG. $N = 500$, $c = 5$. **1:** $t = 10$.
2: $t = 50$. **3:** $t = 100$.

5 The Evolutionary Naming Game

c	$N = 20$	$N = 100$	$N = 200$	$N = 500$
1	1.4 ↓	3.1 ↓	6.2 ↓	16.6 ↓
2	6.7 ↓	13.0 ↓	25.1 ↓	51.4 ↓
3	22.9 ↑	8596.2 ↑	2882.2 ↑	19296.3 ↑
5	3244.1 ↑	11929.9 ↑	73935.8 ↑	176683.8 ↑

Table 5.9: Number of languages (L) for various values of N . $t = 100$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $\rho = 0.03$.

to the number of concepts and agents, with a lower c and N leading to a faster transition.

Note that the last figure, with $N = 500$ and $c = 5$, lacks an image for $t = 200$. This is due to vocabularies growing so large that it is difficult to handle computationally.

5.6 Impact of Parameters

In addition to the experiments described above, several more have been carried out in order to determine the impact of the various parameters. The main question seems to be under which conditions the simulation “converges” into one single language. Section 5.3 shows that with the initial parameters above ($\varepsilon = \sqrt{\frac{10^{-5}}{N}}$, $\rho = 0.03$), this convergence can be seen in all situations where $c \leq 2$. However, $c = 3$ and $c = 4$ have not been tested.

In this section a more systematic approach is followed. Tables 5.9, 5.10, and 5.11 show the number of languages L present in the population after 100 generations (averaged over 9 runs, as in the initial experiments from Section 5.5) for different values of N , ε , and ρ , respectively. Each table shows the situation for the same set of numbers of concepts.

The arrows in the tables denote whether the number of languages is relatively stable (\rightarrow), is increasing (\uparrow), or is decreasing (\downarrow).

Table 5.9 tells us that for the “standard” values of ε and ρ , convergence can be seen when $c \leq 2$, independent of the number of agents N . N does however influence L in a close to proportional manner for larger values of N . However, for small values of N (e.g., $N = 20$) this does not seem to

5.6 Impact of Parameters

c	$\varepsilon = 10^{-5}$	$\varepsilon = \sqrt{\frac{10^{-5}}{N}} = 3.16 \times 10^{-4}$	$\varepsilon = 10^{-4}$
1	8.0 ↓	3.1 ↓	3.7 ↓
2	11.0 →	13.0 ↓	10.1 →
3	4158.9 ↑	8596.2 ↑	9080.4 ↑
5	11860.8 ↑	11929.9 ↑	16094.9 ↑

Table 5.10: Number of languages (L) for various values of ε . $t = 100$, $N = 100$, $\rho = 0.03$.

c	$\rho = 0$	$\rho = 0.001$	$\rho = 0.005$	$\rho = 0.01$	$\rho = 0.03$	$\rho = 0.1$
1	1.6 ↓	1.9 ↓	2.2 ↓	2.2 ↓	3.1 ↓	1.7 ↓
2	2.1 ↓	2.9 ↓	3.8 ↓	6.9 ↓	13.0 ↓	48.9 →
3	2.0 ↓	6 ↑	35.8 ↑	55.0 ↑	3596.2 ↑	110.4 ↑
5	2.8 ↓	17.2 ↑	180.2 ↑	417.8 ↑	3244.1 ↑	210349.2 ↑

Table 5.11: Number of languages (L) for various values of ρ . $t = 100$, $N = 100$, $\varepsilon = \sqrt{\frac{10^{-5}}{N}}$.

be the case.

Table 5.10 shows what happens when ε , i.e., the probability given to the choice of a hearer in addition to its dialog history, is changed. The general trend is that a larger ε means a higher L . A few of the results for a larger number of concepts do not fit this scheme. There is no real reason to believe L begins dropping as ε is increased sufficiently, so this could likely be the result of fluctuations in the simulations. With these quite enormous L values it is easy to imagine large differences between different independent runs.

Another interesting aspect of this table is what happens when $c = 2$. When ε is kept small, i.e., $\varepsilon = 10^{-4}$ or $\varepsilon = 10^{-5}$, the simulation does not seem to converge to a single-language community, but the number of languages instead stabilizes around 10.

The final table, Table 5.11 shows the results for different mutation rates when an agent learns the vocabularies of its parents (ρ). The general trend is doubtlessly that a larger ρ causes a larger number of languages, but there are some exceptions in the data. Most notably, an extreme mutation rate ($\rho = 0.1$) seems to suggest a highly unpredictable behavior. While some

5 The Evolutionary Naming Game

of the rows show a continued larger N for this ρ value, some of them show the *opposite* effect.

The case $\rho = 0$ is worth commenting. In this case all agents learn all words from their parents perfectly, and there is no evolution of words in the world. This leads to a more stable environment — the c value no longer causes the number of languages to grow when high enough, and L in all cases approaches convergence. The absolute number of languages in these cases is higher when c is higher, though.

One last interesting thing to notice is what happens in the case where $c = 2$ and $\rho = 0.1$. As opposed to all other scenarios, this one does not eventually converge into a single-language regime, but instead seems stable around $N = 50$ before and after $t = 100$.

6 Discussion

This chapter discusses the findings of Chapters 4 and 5. Relevant information from the Background Theory (Chapter 2) is also taken into account.

6.1 Evaluation of the NGAWN Implementation

Chapter 4 focused on the implementation of the model described in Lipowska and Lipowski (2012), in this thesis called The Naming Game on Adaptive Weighted Networks (NGAWN).

One of the goals of this part of the research was to be able to verify the model proposed by Lipowska and Lipowski. If that could be done, that model could be used as a basis for the development of other models.

As shown in Section 4.3, the results obtained in this thesis are in most cases close to identical to the Lipowska and Lipowski (2012) results. In particular, all the basic configurations tested yielded a situation where the number of languages in the population decreased as time went by.

As commented in Section 4.3, one noticeable difference between the implementations that can be seen in the plots is a rather small difference in the absolute numbers for success rate in some cases. This fact has not been given any weight when evaluating the implemented model.

In sum, two things can be concluded regarding the NGAWN:

1. Lipowska and Lipowski (2012)'s naming game model and its results can be verified.
2. The implementation done in this thesis is correct and can serve as a skeleton for further experiments.

6.2 Evaluation of the ENG

The model described and implemented in Chapter 5, an extension of NGAWN with evolutionary aspects, was given the name The Evolutionary Naming Game (ENG).

Four reasons for implementing the model were stated in Section 5.3. Below follows a summary of what was achieved in terms of each statement.

Test whether a more realistic model is suitable for research.

As summarized in the introduction of Chapter 5 and further elaborated in Section 5.1, several steps were taken to make ENG more realistic than what was offered by NGAWN. The resulting model indeed proved to be suitable for research. Within the limits of some parameters, it could be seen that the simulations were converging into a single-language regime. Examining the social networks also showed that several distinct communities with strong internal relations were formed.

Find whether the model results in one or several co-existing language communities.

When the number of concepts is small (1 or 2) and the mutation rate ρ is not too large, the model over time converges into a single-language community. This does not mean that no new languages are introduced — as long as the mutation rate is non-zero, new words will arise from time to time. These do not, however, in most instances, form their own communities parallel to the existing one.

A few selected settings gave results where the number of total languages in the population seemed to stabilize. This is not necessarily the same as a number of co-existing language communities, but could also be the result of a rather large number of synonyms per agent.

Test different combinations of parameters to discover their effects.

Section 5.6 dealt with the impact of different parameters on the general situation of the simulation. Table 6.1 summarizes the findings.

Testing setups where the world consisted of a different number of concepts was a major part of the research. While the model certainly “works” for a varying number of concepts, the sizes of the vocabularies seem to

Parameter	Effect of $t \rightarrow \infty$
c	$c \leq 2$ makes the simulation converge to a single-language regime in nearly all instances. $c > 2$ makes the number of synonyms grow drastically.
N	The larger the N , the more generations until convergence.
ε	The larger the ε , the more generations until convergence. <u>Most notable with a small c.</u>
ρ	Always convergence when $\rho = 0$. For $\rho > 0$, the larger the ρ the more generations until convergence. If ρ gets high enough, some normally converging setups might not lead to convergence.

Table 6.1: What effects different values of ENG parameters have as $t \rightarrow \infty$.

quickly get out of hand when $c > 2$. Therefore it can be concluded that using a world with one or two objects seems *more* suitable for research.

The other parameters have proven not to have much of an effect in the general flow of the simulation. They do, however, substantially impact how quickly the simulation reaches towards a given state.

Determine if any configuration of the model compares to aspects of human language evolution. There is at a first glance little that reminds of languages like English or Norwegian when one looks at the output of the ENG simulations. But if we ignore the fact that the complexion that characterizes real languages, some things are also similar to existing languages and their evolution:

- Words (sometimes) change over time. NEFUQ becomes NEXUQ, which becomes NEXUF. In reality sound changes do, however, not happen at random, but instead often follow certain patterns, called sound laws (Anttila, 1989).
- In some configurations, the number of synonyms is kept small and stable.
- Social clusters form, though they are not stable and in most cases seek to converge into one single cluster with a common language.

6 Discussion

Although there is no doubt that the real world experiences a globalization process, this does not mean that all social relations in the world even out.

- The simple language that arises can be compared to the proto-language described by Hurford (2003) (referred in Section 2.1.4), in the sense that there are no syntactic structure. For this to emerge, however, a more sophisticated model is needed.

6.3 Comparison of the Models

Some of the graphs from Chapter 4 can be compared directly to graphs from Chapter 5, as long as one is careful with comparing a single N value (e.g., $N = 500$).

The number of languages in the population, $L(t)$, is not too different between the implementations. ENG drops faster, but when it reaches a specific point the decrease goes much slower, and much more unstable, than what is the case in NGAWN. This can be explained by the presence of mutation, as new words are generated regularly, preventing a quick convergence.

Looking at the success rate, $s(t)$, the implementations seem to differ substantially. In the comparable results from ENG, the success rate quickly reaches almost 1, while this happens much slower in NGAWN.

The share of agents using the most common word, N_1 , is again a little different between the implementations. While NGAWN eventually reaches a state where all agents have the most common word in their vocabulary, ENG does not seem to do so in the comparable plots, even though it reaches a situation where 80–90 % of the agents share the most common word.

Among the things that cannot be directly compared, are the graphs made for $c > 1$ in ENG. It can be seen that this situation without exception makes communication in the world harder, with L increasing and s decreasing, as well as the variance within a single concept increasing — fewer agent share the same words.

All in all, 5 in many ways seems more unstable than 1 in most aspects. It seems likely that the mutation of words has a large part of the

6.3 *Comparison of the Models*

responsibility for this. New words are generated every generation, making convergence go slower and less smooth. Even when the simulation is converging into a single-language community, new words will still pop up, though most of them at that stage will be short-lived.

7 Conclusion and Future Work

7.1 Conclusions

This section will make some conclusions based on the goal and research questions from the Introduction chapter (Chapter 1).

7.1.1 Research Questions

The research questions posed in Section 1.1 are here revisited, and an evaluation of them is made.

Research question 1 *Is The Naming Game on Adaptive Weighted Networks (NGAWN), proposed by Lipowska and Lipowski (2012), verifiable and reproducible?*

This question has an easy answer: Yes, the model is verifiable. As discussed in Section 6.1, the results obtained are in most cases close to identical. This made it possible to use this implementation as a foundation for the The Evolutionary Naming Game (ENG) implementation.

Research question 2 *Can an evolutionary model based on NGAWN be constructed?*

An evolutionary model in the naming game framework has been constructed. It does not incorporate a proper Evolutionary Algorithms (EA), but uses several elements known from there: Generations with replacement of agents, reproduction, inheritance, and some kind of mutation.

Research question 3 *What can be achieved in terms of realistic simulations with an evolutionary model based on NGAWN?*

7 Conclusion and Future Work

As discussed in Section 6.2, ENG offer some interesting qualities in terms of realism in the setup. It is, however, important to note that it is still a simple computation model taking inspiration from the real world, and not in any way a model simulating the real world.

As has already been mentioned, the languages emerging from the ENG have more in common with the assumed proto-language than modern human languages. And as was stated in 2.1.4, this is in line with the typical computation approach — begin from a community without any language and run simulations to create a more advanced language as time goes by.

ENG does not support any further complication of the language (e.g., syntactic structure), and running it for a large number of generation would in most cases not yield any results comparable to the language situation in the world, as close to no configurations maintain a stable environment over time, neither quickly converging to a single-language regime nor producing an ever increasing number of synonyms.

7.1.2 Goal

The goal stated in Section 1.1 was:

Goal *Exploring in which ways an evolutionary extension of a naming game model can be used as a simplified model of how languages evolve over time.*

An extension of NGAWN has been made, several configurations have been tested, and the results have been presented. The results have been discussed in relation to the implementation of NGAWN, and in relation to language evolution in the real world.

Some configurations proved unstable in the sense that the number of words in the simulation became unmanageable within a relatively small number of generations. Others produced situations where a single-language regime with all agents having high social relations between each other. A selected few configurations lead to situations that seemed stable in the number of total languages, but these will need to be examined more closely to determine if this is the case also with a large number of generations (more than the 1000 that were tested in those particular cases).

There are, however, certain aspects of the simulations that resemble human language in a way NGAWN could not. These are summarized in Section 6.2.

7.2 Future Work

There are many areas of possible further research within both naming games and the rest of the field of computational linguistics. This section will focus on some ideas for future work more or less in direct relation to the work done in this thesis.

There are doubtlessly some problems with the proposed model. One of the most apparent ones is the abrupt increase in the size of agent vocabularies in certain configurations, especially when there are three or more concepts in the world. An interesting technique to tackle this development could be to involve some kind of alignment scheme as the one described in “the naming game with alignment” (Steels and Loetzsch, 2012). This would mean to associate each word in an agent’s vocabulary with a numerical score, making words which are not communicated successfully with inactive, i.e., unavailable for use.

Another interesting area of work, particularly if a more stable state could be developed, for example using a technique like the one described above, could be to run the simulations for a larger number of generations. There are two main reasons this has not been done in this thesis:

1. In many cases it is not very interesting, as it is quickly apparent if the trend is convergence or divergence.
2. The model is quite computationally demanding, and with the current setup many configurations cannot be run for a large number of generations without extensive computational power.

If one wishes to develop the model in a more realistic direction, one idea could be to introduce social roles. For example in (Munroe and Cangelosi, 2002) agents can take the role of teacher or speaker, depending on their age. This is one of many imaginable steps that can be taken to make the social aspects of ENG more similar to the culture surrounding human language learning.

7 Conclusion and Future Work

More steps can also be taken to obtain a model with a stronger evolutionary profile. As done in (Lekvam, 2014), it is possible to equip the agents with one or more genes governing their behavior in certain stages of the simulations, for example which agents they choose to communicate with.

Bibliography

- Dorota Lipowska and Adam Lipowski. Naming game on adaptive weighted networks. *Artificial life*, 18(3):311–323, 2012.
- Chris Knight, Michael Studdert-Kennedy, and James R. Hurford. Language: A Darwinian adaptation? In Chris Knight, Michael Studdert-Kennedy, and James R. Hurford, editors, *The Evolutionary Emergence of Language: Social Function and the Origins of Linguistic Form*, chapter 1, pages 1–15. Cambridge University Press, Cambridge, 2000.
- Luc Steels. Modeling the cultural evolution of language. *Physics of Life Reviews*, 8(4):339–356, December 2011.
- Theresa Schilhab, Frederik Stjernfelt, and Terrence Deacon. Introduction — searching the missing links. In Theresa Schilhab, Frederik Stjernfelt, and Terrence Deacon, editors, *The Symbolic Species Evolved*, chapter 1, pages 1–5. Springer, Dordrecht/Heidelberg/London/New York, 2012.
- J Mark Baldwin. A new factor in evolution. *The american naturalist*, 30(354):441–451, 1896.
- Stefano Nolfi and Marco Mirolli. A synthetic approach to the study of the evolution of communication and language. In Stefano Nolfi and Marco Mirolli, editors, *Evolution of Communication and Language in Embodied Agents*, chapter 1, pages 1–9. Springer, Berlin/Heidelberg, 2010.
- Cristopher Routledge and Siobhan Chapman. *Key Ideas in Linguistics and the Philosophy of Language*. Edinburgh University Press, Edinburgh, 2009.
- Luc Steels. Self-organization and selection in cultural language evolution. In Luc Steels, editor, *Advances in Interaction Studies: Experiments in*

Bibliography

- Cultural Language Evolution*, chapter 1, pages 1–37. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2012.
- Ludwig Wittgenstein. 1958. *Philosophical investigations*, 3, 1953.
- Luc Steels and Martin Loetzsch. The grounded naming game. In Luc Steels, editor, *Advances in Interaction Studies: Experiments in Cultural Language Evolution*, chapter 2, pages 41–59. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2012.
- Derek Bickerton. *Language & Species*. The University of Chicago Press, Chicago/London, 1990.
- Remi van Trijp. The evolution of case systems for marking event structure. In Luc Steels, editor, *Advances in Interaction Studies: Experiments in Cultural Language Evolution*, chapter 7, pages 169–205. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2012.
- Luciano da Fontoura Costa, Osvaldo N Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiquiera, Matheus Palhares Viana, and Luis Enrique Correa Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, 2011.
- Andrea Baronchelli, Luca Dall’Asta, Alain Barrat, and Vittorio Loreto. Bootstrapping communication in language games. In *The Evolution of Language, Proceedings of the 6th International Conference (EVOLANG6)*. World Scientific Publishing Company, 2006.
- Tao Gong, Jinyun Ke, James W Minett, and William SY Wang. A computational framework to simulate the coevolution of language and social structure. In *Artificial Life IX: Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems*, pages 158–64, 2004.
- Torvald Lekvam. Co-evolving language and social structure using a genetic algorithm, 2014.
- Derek Bickerton. Language evolution: A brief guide for linguists. *Lingua*, 117(3):510–526, 2007.

- James R Hurford. The language mosaic and its evolution. *Studies in the Evolution of Language*, 3:38–57, 2003.
- Daniel Ashlock. *Evolutionary Computation for Modeling and Optimization*. Springer, New York, 2006.
- Eörs Szathmáry. Evolution of language as one of the major evolutionary transitions. In Stefano Nolfi and Marco Mirolli, editors, *Evolution of Communication and Language in Embodied Agents*, chapter 3, pages 37–53. Springer, Berlin/Heidelberg, 2010.
- Vittorio Loreto. Theoretical tools in modeling communication and language dynamics. In Stefano Nolfi and Marco Mirolli, editors, *Evolution of Communication and Language in Embodied Agents*, chapter 5, pages 67–81. Springer, Berlin/Heidelberg, 2010.
- Adam Lipowski and Dorota Lipowska. Bio-linguistic transition and Baldwin effect in an evolutionary naming-game model. *International Journal of Modern Physics C*, 19(3):399–407, March 2008.
- Dorota Lipowska. Naming game and computational modelling of language evolution. *Computational Methods in Science and Technology*, 17(1-2): 41–51, 2011.
- Torvald Lekvam, Björn Gambäck, and Lars Bungum. Agent-based modeling of language evolution. In *Proc. of 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)@ EACL*, pages 49–54, 2014.
- Derek Bickerton. Two neglected factors in language evolution. In *The Evolution of Language*, pages 26–33. World Scientific, 2008.
- Steven Pinker. *The language instinct*. New York: HarperCollins, 1994.
- Michał B Paradowski and Aleksandra Bator. Perceived effectiveness of language acquisition in the process of multilingual upbringing by parents of different nationalities. *International Journal of Bilingual Education and Bilingualism*, pages 1–19, 2016.
- Raimo Anttila. *Historical and comparative linguistics*, volume 6. John Benjamins Publishing, 1989.

Bibliography

Vittorio Loreto, Andrea Baronchelli, Animesh Mukherjee, Andrea Puglisi, and Francesca Tria. Statistical physics of language dynamics. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(04):P04006, 2011.

Steve Munroe and Angelo Cangelosi. Learning and the evolution of language: the role of cultural variation and learning costs in the baldwin effect. *Artificial Life*, 8(4):311–339, 2002.