Active control of complexity growth in Language Games

William Schueller

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

**DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX**

ÉCOLE DOCTORALE DE
MATHÉMATIQUES ET INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

Par William Schueller

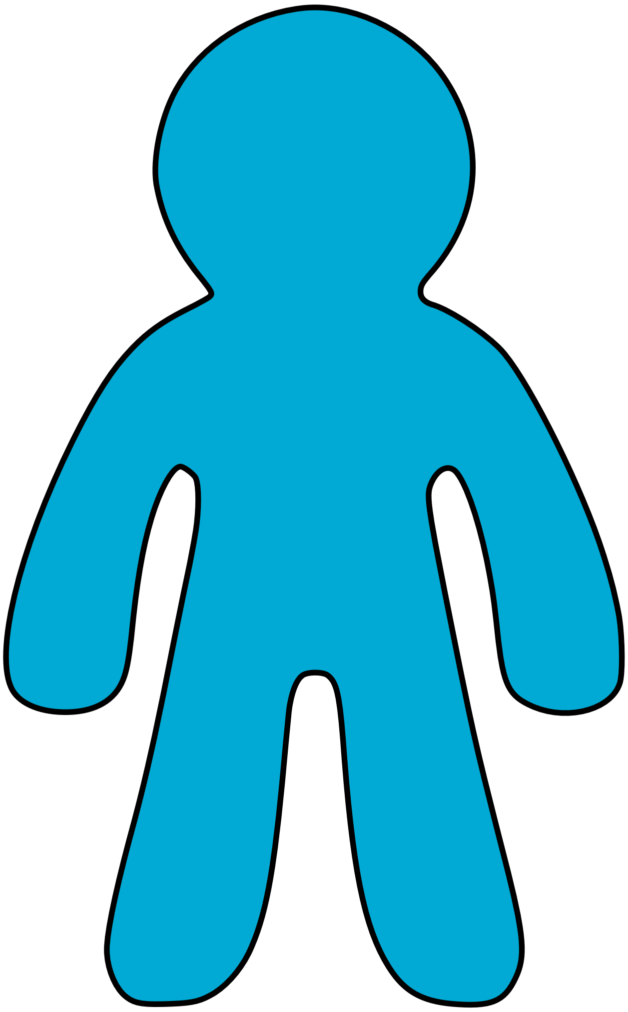
Active Control of Complexity Growth in Language Games

Sous la direction de : Pierre-Yves Oudeyer

Date de soutenance : 10 décembre 2018

Membres du jury :

Bart de Boer Président du jury/Rapporteur
Kenny Smith Rapporteur
Dan Dediu Examinateur
Andrea Baronchelli Examinateur
Vittorio Loreto Invité
Pierre-Yves Oudeyer Directeur de thèse



Acknowledgments

Acknowledgments in other languages (parts in Français - Türkçe - Català - Italiano - Español) can be found after this English-only version.

I first would like to thank the members of the jury, it has been an honor to have you all to read my work and come to the defense. Your work has been an inspiration, and the chats with you and your many students at various conferences and events. Thank you Kerem for introducing me to many of those students, and thank you Ramon for not only bringing me into the Evolang community in the first place, but also for motivating me to learn catalan while doing castells.

I would like to thank my supervisor Pierre-Yves Oudeyer, without whom this work would not have been possible. He taught me to aim at feasible goals and let perfection aside, because – as would say another person who brought incredible help and mentoring, Fabien Benureau – *optimal isn't good enough*. I thank all members of the Flowers team for the time I shared with them, especially Benjamin the writing buddy, Baptiste the one who appreciates my puns, Flo the vending machine hacker, Didier the joyful loud speaker, Alexandra the model of tenacity, Hélène the motivation coach, Adrien the honest Rocket League commentator, and last but not least Théo, a skilled expert at everything – maybe just not catching mice.

Many other people at Inria also played an important part during this thesis. Thanks to Fabien, Xavier, Bhargav, Philippe, Sol (who efficiently saved me 2 days before the defense), Pierre-Antoine, Jonathan, Alex Spriet (who hosted me in NYC before a deadline), Marie, Cynthia. Of course, I need to mention Marc Fuentes, with whom I could share my love for puns and languages.

Special thanks to Sandy and Atlal who made it possible to build the second version of the experiment.

And thanks to the administrative assistants, especially Luce and Nicolas who seem to have magical skills for paperwork.

My stay in Rome was extremely helpful to refocus on the thesis, and I thank a lot Vittorio for the opportunity he gave me to work at the Sapienza for half a year, and for the efficient discussions we had. Of course, I need to thank Miguel for his patience to slowly tear my spanish towards a functional italian, and for his everyday chats. I also thank Pietro who painfully heard my first italian puns, Betta the nerd who helped me a lot, Vito and Bernardo who I saw only lately during my stay. This stay in Rome was also a great time thanks to my friends, Fra', Michele, Paola, Estelle, Rachel, Maghetto, Vieri, Marica, and many others. I hope to see all soon to go eat some suppli'!

Thanks to all my friends; Désirée for the gardening sessions at the castle, Yan for the paraglider-camping parties, Bruno for hosting me at the beginning of my stay, Olivier for showing up every time from another country, Thomas for the useful discussions, and Tianhan, Vincent and Lucas for their great help.

Thanks to all the friends who helped me not to forget my turkish! Fatih and Oğuz as colleagues, Dilan, Gisèle, Éleonore, Günay and others in Bordeaux, Fatih, Toprak, Özge for their visits, and of course Marie-Christine, Ali and Çöp Şiş for their hospitality each time I would go back to İstanbul; and my many friends there, Nico, Mina, Didem, Ilyas, Ruhi, Jean-Mi, Magali, and all the colleagues at GSU.

Thanks to my grandma for her unconditional love and many boxes full of kugelhopfs, pretla and schnaps-filled chocolate! Thanks to my mom and my cousins for their support and understanding of my sparse presence and poor organisation over the last years. Thanks to my cousin Vincent for introducing me at a young age to the world of research.

Of course, none of this would have been possible without my many flatmates (more than 20 over the course of the PhD), especially Laura, Noémie, Zazoo, Isa. The constant over the PhD years was Seb and Leana, who became family to my eyes; as well as Jad, who sadly had to leave France, we miss you!

And last but not least, thanks to Alex and Pauline for your friendship and support, I learned so much from you both, and I am looking forward to attending your own defenses!

Remerciements

Je commence par remercier les membres du jury, c'était un honneur que vous lisiez mon travail et veniez pour la défense. Votre travail a été une inspiration, ainsi que les discussions avec vous et vos nombreux étudiants au cours des différents évènements et conférences qui ont ponctués ma thèse. Kerem, beni çok kişilere tanıştırdığını için teşekkür ediyorum. Graciès a Ramon per l'introducció en la comunitat Evolang, i també per motivar-me a aprendre el català fent castells.

Merci à mon directeur Pierre-Yves Oudeyer, sans qui ce travail n'aurait pas été possible. Il m'a appris à me fixer des buts faisables au lieu de toujours viser la perfection, puisque – comme le dirait une autre personne ayant apporté une aide et un mentoring conséquents, Fabien Benureau – *optimal isn't good enough*. Merci à tous les membres de l'équipe Flowers, particulièrement Benjamin le compagnon d'écriture, Baptiste qui apprécie mes jeux de mots, Flo der Automatenhacker, Didier le joyeux haut-parleur, Alexandra le modèle de ténacité, Hélène la coach motivation, Adrien le très honnête commentateur de Rocket League, et pour finir Théo, l'expert en tout – sauf attraper les souris.

De nombreuses autres personnes à Inria ont joué un rôle important au cours de cette thèse. Merci à Fabien, Bhargav, Philippe, Sol (quién me salvó dos días antes de la presentación), Pierre-Antoine, Jonathan, Damien, Alex Spriet (qui m'a hébergé à NYC juste avant une deadline), Marie, Cynthia. Marc, eskerrik asko, j'espère pouvoir bientôt te raconter des jeux de mots en euskara.

Merci particulièrement à Sandy et Atlal qui ont rendu possible la mise en place de la deuxième version de l'expérience.

Et merci aux assistants et assistantes administratifs, particulièrement Luce et Nicolas, qui semblent être des magiciens de la bureaucratie.

Il mio soggiorno a Roma mi ha aiutato per riconcentrarmi sulla significazione della tesi, ringrazio molto Vittorio per l'opportunità che mi ha dato di lavorare nella Sapienza e per le discuzione interesante ed efficace. *Tengo que* ringraziare Miguel per la sua pazienza cambiando lentamente il mio *castellano* in italiano, e anche per le discuzione ogni giorno. Grazie Pietro, lo so che non era facile ascoltare alle mie prime battute in italiano, grazie a Betta la nerd che anche mi ha aiutato molto, Vito e Berna', pure se non ci siamo visti molto in Roma. Questo soggiorno a Roma era anche un buon tempo grazie a tutti gli amici, Fra', Michele, Paola, Estelle, Rachel, Maghetto, Vieri, Marica, e molti altri. Spero di vedervi presto per magna' suppli'!

Merci à tous les amis: Désirée pour les séances de jardinage au château, Yan pour les soirées parachute, Bruno pour m'avoir accueilli et présenté Bordeaux, Olivier qui revenait chaque fois d'un autre pays, Thomas pour les disscussions qui se sont révélées très utiles, et Tianhan, Vincent et Lucas pour leur aide décisive au moment opportun.

Birçok arkadaşlarımın sayesinde Türkçemi unutmadım! Fatih ve Oğuz işarkadaşlar olarak; Dilan, Gisèle, Éleonore, Günay ve başka arkadaşlar Bordo'da; Fatih, Toprak, Özge misafir olarak, ve tabii Marie-Christine, Ali ve Çöp Şiş İstanbul'a döndüğümlerde; bir de oradaki arkadaşlar: Nico, Mina, Didem, Ilyas, Ruhi, Olcay, Jean-Mi, Magali, Fred ve GSÜ'nde işarkadaşlar.

Merci Mamie pour ton rire et ton affection, mais aussi pour les nombreuses boîtes remplies de kugelhopfs, pretla et chocolats au schnaps! Je remercie ma mère et mes cousins pour leur soutien et leur compréhension de mon faible taux de présence ces derniers temps, et mon cousin Vincent pour m'avoir présenté très tôt le monde de la recherche.

Bien sûr, tout cela n'aurait jamais été possible sans l'aide et le soutien de mes nombreux colocataires (plus de 20 durant la thèse), particulièrement Laura, Noémie, Zazoo et Isa. Les éléments constants tout au long de ces années de thèse ont été Seb et Leana, qui sont devenus une sorte de nouvelle famille; sans oublier Jad, qui a malheureusement du quitter la France quelques mois avant la soutenance.

Merci à Alex et Pauline pour votre soutien, votre amitié, et tout ce que vous m'avez appris. Et maintenant, j'attends de vous voir défendre à votre tour! (Ça vaut pour toi aussi, Leana!)

# Abstract

Keywords: active learning – self-organization – language emergence – language evolution

Social conventions are learned mostly at a young age, but are quite different from other domains, like for example sensorimotor skills. The first people to define conventions just picked an arbitrary alternative between several options: a side of the road to drive on, the design of an electric plug, or inventing a new word. Because of this, while setting a new convention in a population of interacting individuals, many competing options can arise, and lead to a situation of growing complexity if many parallel inventions happen. How do we deal with this issue?

Humans often exhert an active control on their learning situation, by for example selecting activities that are neither too complex nor too simple. This behavior, in cases like sensorimotor learning, has been shown to help learn faster, better, and with fewer examples. Could such mechanisms also have an impact on the negotiation of social conventions?

A particular example of social convention is the lexicon: which words we associated with given meanings. Computational models of language emergence, called the Language Games, showed that it is possible for a population of agents to build a common language through only pairwise interactions. In particular, the Naming Game model focuses on the formation of the lexicon mapping words and meanings, and shows a typical burst of complexity before starting to discard options and find a final consensus.

In this thesis, we introduce the idea of active learning and active control of complexity growth in the Naming Game, in the form of a topic choice policy: agents can choose the meaning they want to talk about in each interaction. Several strategies were introduced, and have a different impact on both the time needed to converge to a consensus and the amount of memory needed by individual agents.

Firstly, we artificially constrain the memory of agents to avoid the local complexity burst. A few strategies are presented, some of which can have similar convergence speed as in the standard case. Secondly, we formalize what agents need to optimize, based on a representation of the average state of the population. A couple of strategies inspired by this notion help keep the memory usage low without having constraints, but also result in a faster convergence process.

We then show that the obtained dynamics are close to an optimal behavior, expressed analytically as a lower bound to convergence time.

Eventually, we designed an online user experiment to collect data on how humans would behave in the same model, which shows that they do have an active topic choice policy, and do not choose randomly.

Contributions from this thesis also include a classification of the existing Naming Game models and an open-source framework to simulate them.

Abstract

Mots-Clés: apprentissage actif – auto-organisation – émergence du langage – évolution du langage

Nous apprenons très jeunes une quantité de règles nous permettant d'interagir avec d'autres personnes: des conventions sociales. Elles diffèrent des autres types d'apprentissage dans le sens où les premières personnes à les avoir utilisées n'ont fait qu'un choix arbitraire parmi plusieurs alternatives possibles: le côté de la route où conduire, la forme d'une prise électrique, ou inventer de nouveaux mots. À cause de celà, lorsqu'une nouvelle convention se crée au sein d'une population d'individus interagissant entre eux, de nombreuses alternatives peuvent apparaître et conduire à une situation complexe où plusieurs conventions équivalentes coexistent en compétition. Il peut devenir difficile de les retenir toutes, comment faisons-nous pour trouver un accord efficacement?

Nous exerçons communément un contrôle actif sur nos situations d'apprentissage, en par exemple sélectionnant des activités qui ne soient ni trop simples ni trop complexes. Il a été montré que ce type de comportement, dans des cas comme l'apprentissage sensori-moteur, aide à apprendre mieux, plus vite, et avec moins d'exemples. Est-ce que de tels mécanismes pourraient aussi influencer la négociation de conventions sociales?

Le lexique est un exemple particulier de convention sociale: quels mots associer avec tel objet ou tel sens? Une classe de modèles computationels, les Language Games, montrent qu'il est possible pour une population d'individus de construire un langage commun via une série d'interactions par paires. En particulier, le modèle appelé Naming Game met l'accent sur la formation du lexique reliant mots et sens, et montre une typique explosion de la complexité avant de commencer à écarter les conventions synonymes ou homonymes et arriver à un consensus.

Dans cette thèse, nous introduisons l'idée de l'apprentissage actif et du contrôle actif de la croissance de la complexité dans le Naming Game, sous la forme d'une politique de choix du sujet de conversation, applicable à chaque interaction. Différentes stratégies sont introduites, et ont des impacts différents sur à la fois le temps nécessaire pour converger vers un consensus et la quantité de mémoire nécessaire à chaque individu.

Premièrement, nous limitons artificiellement la mémoire des agents pour éviter l'explosion de complexité locale. Quelques stratégies sont présentées, certaines ayant des propriétés similaires au cas standard en termes de temps de convergence. Dans un deuxième temps, nous formalisons ce que les agents doivent optimiser, en se basant sur une représentation de l'état moyen de la population. Deux stratégies inspirées de cette notion permettent de limiter les besoins en mémoire sans avoir à contraindre le système, et en prime permettent de converger plus rapidement.

Nous montrons ensuite que la dynamique obtenue est proche d'un comportement théorique optimal, exprimé comme une borne inférieure au temps de convergence.

Finalement, nous avons mis en place une expérience utilisateur en ligne sous forme de jeu pour collecter des données sur le comportement d'utilisateurs réels placés dans le cadre du

modèle. Les résultats suggèrent qu'ils ont effectivement une politique active de choix de sujet de conversation, en comparaison avec un choix aléatoire.

Les contributions de ce travail de thèse incluent aussi une classification des modèles de Naming Games existants, et un cadriciel open-source pour les simuler.

Contents

List of Figures

List of Tables

List of Algorithms

Résumé en Français

L'enfance est remplie de nombreux défis, l'un des plus importants étant d'apprendre et acquérir une énorme quantité de compétences et de connaissances; tout cela en un temps limité et à un rythme impressionnant. Les enfants en bas âge apprennent rapidement à maîtriser leurs mouvements et leur corps, à prononcer des mots, et à interagir avec différents objets et avec d'autres personnes; plus généralement avec le monde qui les entoure.

Ils passent d'une activité à l'autre et sont soumis à un flux constant de nouvelles informations. Ils s'intéressent rapidement à quelque chose, mais s'ennuient aussi facilement et passent à autre chose.

Mais comment décident-ils des nouvelles activités à choisir ? Leur motivation dans ces exemples n'est pas externe, mais *intrinsèque* (Berlyne, 1960; Ryan and Deci, 2000).

Les théories développées pour définir les activités intrinsèquement motivantes incluent le concept de *Flow* (Csikszentmihalyi, 1991) : les activités trop difficiles peuvent être source d'anxiété, et celles trop simples source d'ennui. Dans un contexte d'apprentissage, les nouvelles compétences qui sont acquises deviennent progressivement plus faciles. D'autres tâches, au contraire, étaient peut-être trop complexes au début de l'apprentissage, mais sont maintenant accessibles, grâce aux compétences nouvellement acquises. Ce processus fait que la zone de *Flow* optimale se déplace progressivement dans le temps (voir figure).

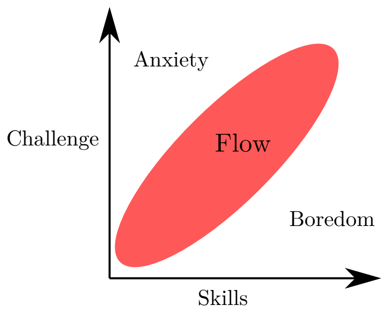


Figure 1: Illustration simplifiée du *flow* (Csikszentmihalyi, 1991) : avec un certain niveau de compétences, les défis peuvent être trop complexes et causer de l'anxiété, trop simples et de l'ennui. Le *flow* est une zone intermédiaire, où l'apprentissage peut être plus efficace.

Cette idée d'état intermédiaire se retrouve dans d'autres théories. Dans Berlyne, 1965, il a été avancé que beaucoup de termes associés à la motivation intrinsèque (*incongruité*, *complexité*, *nouveauté*, ...) pourraient être décrits comme des variables inspirées de la théorie de l'information. Différentes fonctions peuvent être considérées comme une récompense intrinsèque et ont été examinées dans Oudeyer and Kaplan, 2009.

Une alternative à une potentielle mesure des récompenses intrinsèques dans le cerveau, qui peut être complexe et invasive, est d'envisager les robots, et plus généralement les algorithmes d'apprentissage. Comme les bébés, les robots interagissent avec leur environnement par l'intermédiaire d'un corps dont les nombreux actionneurs (moteurs ou muscles) sont complexes à maîtriser.

Même des tâches que nous considérons naturellement très simples, parce que nous les exécutons dans notre vie quotidienne sans même y penser, sont en réalité très difficiles. Par exemple, la marche est plus difficile qu'on ne l'imagine et implique un équilibre subtil et une coordination entre les muscles. Les corps des robots sont généralement conçus pour permettre aux ingénieurs de calculer des modèles et des paramètres efficaces pour la marche (moteurs plus puissants, centre de gravité plus bas, forme des jambes, etc. – voir figure pour une illustration). Au contraire, un enfant apprend en un an à se lever et à marcher, à découvrir son corps et s'y adapter, car il change et grandit rapidement en même temps. L'enfant exerce un contrôle actif sur la croissance de la complexité de ses interactions avec son environnement, ce qui lui permet d'apprendre plus efficacement.

Les comportements inspirés des théories sur la *motivation intrinsèque* peuvent être implémentés comme algorithmes et utilisés en particulier dans les robots : par exemple dans l'expérience *Playground*. (Oudeyer and Kaplan, 2006), un robot est placé dans une aire de jeux pour bébé et apprend activement à interagir avec son environnement, en décidant quelles activités sont plus intéressantes que d'autres, suivant une mesure de progrès d'apprentissage. Il y a deux facettes complémentaires à ce type d'approche, appelée *developpementale* : les expériences robotiques qui permettent de mieux comprendre le comportement de l'enfant en reproduisant les modèles associés (activités de commutation, voies de développement, etc.), et l'adaptation des connaissances en psychologie pour concevoir des algorithmes plus efficaces d'apprentissage machine et d'intelligence artificielle en général, les tester sur robots ou modèles computationnels.

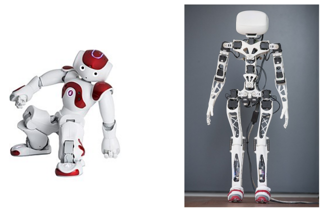


Figure 2: Le robot Nao (SoftBanks Robotics) est un robot commercial, dont les jambes et les pieds ont été conçus pour lui permettre de marcher facilement. Gauche : Le robot humanoïde Poppy a été conçu pour étudier la marche humaine, et a des proportions corporelles et des forces motrices semblables à celles d'un enfant, mais ne peut pas encore marcher sans assistance (Lapeyre et al., 2014).

Figure 3: L'expérience du terrain de jeux ou *Playground* (Oudeyer and Kaplan, 2006) : un robot choisit ses activités au cours du temps, en fonction de ses progrès d'apprentissage. L'activité comprend la saisie et le toucher d'objets, ainsi que la communication avec un pair.

Les humains ne font pas qu'apprendre individuellement, mais peuvent aussi apprendre à se coordonner collectivement; par exemple lors de mouvements de groupe, d'exploration commune, ou de consensus sur des conventions sociales, comme par exemple le langage.

La langue est généralement enseignée à l'école sous la forme d'un ensemble statique de règles et de définitions, comprenant souvent de longues listes d'exceptions et d'irrégularités. Ces règles peuvent cependant faire l'objet de variations et de suppression/création de règles, non pas parce qu'une autorité comme l'Académie française le décide, mais parce que les usagers de la langue changent spontanément et inconsciemment leurs habitudes. De nouveaux mots et de nouvelles règles peuvent être transmis à d'autres orateurs qui, à leur tour, les propageront et les modifieront davantage, ou refuseront de les adopter et de s'en tenir à leurs propres règles. Les règles peuvent se reproduire et se propager, mais elles sont aussi en concurrence les unes avec les autres pour au moins une ressource limitée : les utilisateurs de ces langues. En d'autres termes, la langue est soumise à un processus évolutif.

Si l'évolution biologique peut être liée à certains aspects du langage (par exemple la forme du larynx ou des parties spécifiques du cerveau), elle ne peut expliquer l'évolution rapide du langage

: l'usage d'un nouveau mot peut facilement se propager dans le monde en quelques heures. Cela révèle également un autre aspect de l'évolution du langage qui contredit l'héritage génétique classique de l'information : le langage peut être transmis directement d'une personne à l'autre, ou *horizontalement*, par opposition à la transmission *verticale* d'une génération à l'autre. Le langage peut être décrit comme sujet à l'*évolution culturelle*, et résulte des interactions entre les individus.

Le terme de language games a été introduit par Wittgenstein, 1953 pour décrire bon nombre de nos interactions quotidiennes qui peuvent soit utiliser le langage, soit constituer un substrat communicatif pour l'émergence du langage. Dans les modèles informatiques de l'évolution du langage, il fait référence à une classe de modèles multi-agents (Steels, 1995) dans laquelle le langage est considéré comme un système complexe adaptatif (Steels, 2000). Ils se concentrent sur les propriétés d'auto-organisation résultant de la transmission horizontale du langage, et montrent de façon intéressante que certaines caractéristiques du langage peuvent émerger spontanément dans une population d'agents, sans un contrôle centralisé ou un langage partiellement inné.

L'une des versions les plus simples s'appelle le *Naming Game* : les agents s'accordent progressivement sur un lexique, faisant correspondre des objets ou concepts à des mots (Loreto, Baronchelli, et al., 2011; Steels, 1995; Wellens, 2012). Les agents interagissent par paires choisies au hasard, de manière décentralisée; mais parviennent à converger vers un lexique commun à tous après un certain nombre d'interactions. Ce modèle étant au cœur de cette thèse, le chapitre 2 est consacré à sa description détaillée.

Une approche au contrôle actif de la croissance de la complexité a été introduite par Oudeyer and Delaunay, 2008, reposant sur le fait que moins de connaissances sont pré-partagées par les agents : un choix actif du sujet de chaque communication (parmi les différents objets ou concepts possibles). Lorsqu'il s'agit de nommer plusieurs objets, l'accord peut être obtenu plus rapidement si les agents se contentent d'abord des étiquettes de certains objets, puis passent à en nommer d'autres. Cette approche sera l'élément central de cette thèse.

Cette thèse utilise donc un modèle computationnel multi-agent mettant en jeu une dynamique collective, le Naming Game, et une approche spécifique à l'introduction d' un contrôle actif de la croissance de la complexité: le choix actif du sujet de la communication (Oudeyer and Delaunay, 2008). Nous nous concentrons sur deux propriétés des stratégies qui en découlent : un accord global peut être construit plus rapidement, et moins de mémoire est nécessaire pour y arriver.

Le chapitre 2 présente le modèle et les nombreuses variantes qui ont été étudiées, ainsi que les mesures et propriétés pertinentes du modèle. Pour rester plausible sur le plan cognitif, la mémoire nécessaire doit rester faible et ne pas croître en fonction de la taille de la population; ce qui n'est pas le cas dans la version standard du Naming Game.

Le chapitre 3 introduit la notion de choix de sujet actif, et présente quelques stratégies sous contrainte de mémoire: les homonymes et synonymes sont interdits. Les stratégies Information Gain Maximization et Chunks reposent uniquement sur la taille actuelle du lexique de l'agent. Les stratégies Minimal Counts et Success Threshold reposent sur un comptage des succès et des échecs parmi les tentatives de communication, respectivement d'une manière absolue ou relative. Nous montrons que les deux dernières stratégies permettent de converger beaucoup plus rapidement qu'une politique de choix aléatoire des sujets de communication.

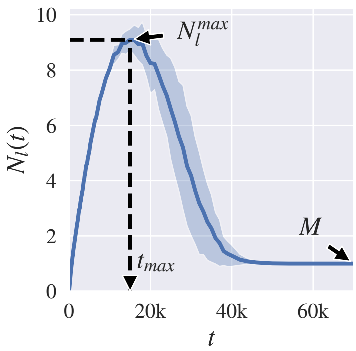


Figure 4: Évolution typique du nombre de mots (mémoire) par agent. Le système passe par un maximum, et c'est sur cette croissance de la complexité que les algorithmes introduits dans cette thèse tenteront d'exercer un contrôle.

Le chapitre 4 reconsidère l'homonymie et la synonymie : l'objectif est de concevoir de nouvelles stratégies qui contrôleront elles-mêmes l'utilisation de la mémoire. Une nouvelle mesure est introduite, la mesure LAPS, ou Probabilité Locale Approximative de Succès. Il s'agit d'une estimation à l'échelle de l'agent du niveau d'accord avec l'ensemble de la population. Une stratégie maximisant cette mesure est présentée, et montre des propriétés remarquables en termes de temps de convergence et d'utilisation de mémoire. La stratégie LAPSmax n'utilise cependant pas la maximisation directe de la mesure LAPS, mais une heuristique : il serait trop difficile de calculer autrement. Une mesure plus simple, proche de LAPS et appelée cohérence, conduit à une autre stratégie avec maximisation directe. Cette stratégie converge encore plus vite que LAPSmax, mais utilise un peu plus de mémoire. Ces deux stratégies sont robustes à des contraintes même fortes sur le nombre de mots possibles.

Le chapitre 5 présente des estimations statistiques de certaines des mesures. En particulier, nous calculons une borne inférieure statistique au temps de convergence. Nous utilisons ces expressions pour définir six mesures de performance, caractérisant le temps de convergence et l'utilisation de la mémoire. Nous comparons les stratégies des chapitres précédents et montrons que la stratégie de cohérence est presque optimale.

Le chapitre 6 présente une expérience utilisateur qui a été menée pour observer le comportement de participants réels dans le contexte d'un Naming Game, en tant qu'application web en ligne. Comme nous ne nous intéressons qu'aux décisions des agents, nous pouvons simuler le reste de la population et mener facilement l'expérience avec des participants individuels. Les comportements observés montrent moins d'exploration qu'une politique de choix aléatoire du sujet, et suggèrent donc que les humains exercent un contrôle actif dans ce cadre.

En conclusion, nous montrons donc qu'un contrôle actif de la croissance de la complexité peut se montrer très efficace dans le cadre d'une dynamique collective, à travers l'exemple particulier de l'émergence du lexique. Nous exhibons une variété de stratégies possibles, certaines d'entre elles étant quasi-optimales. L'expérience utilisateur suggère que les humains utilisent naturellement ce type de comportement.

Inria Bordeaux Sud-Ouest

Part I

Introduction

ICDL-Epirob Conference, Providence, RI. August 2015

1 Introduction

Contents

1.1 Social conventions

Throughout our life, we never stop learning. At an early age, we learn to grab objects, to walk, or to produce sounds. We later learn to write, speak, play games, and more generally interact with others. There is however an important distinction between those two lists: The elements of the first are universals, while elements of the other are not. If a child learns to walk in a given country, they will still be able to use this knowledge at the other side of the planet. On the contrary, the local language would probably be different, people may write in the opposite direction, or drive on the other side of the road. This is because they are social conventions: there is a need to agree on how we interact with each other, to follow common rules, but those rules themselves are arbitrary and did not exist before our societies.

Nevertheless, we perceive both types of learning in the same way, as we learn social conventions from our parents, teachers, or supervisors. They seem to be determined before we learn them, to pre-exist ourselves. But generation after generation, we keep on modifying them and creating new ones: we would surely not be able to understand the inhabitants of our own city from a few centuries ago, and our own parents probably do not understand some parts of our behaviors. Because of their arbitrary nature, the collective *consensus* on those conventions is a constant negotiation.

Along with a pressure to keep previously agreed conventions, there remains an exploration of other options by new generations or individuals joining the group. There might be conventions that are more efficient than others, which would naturally be selected and preferred over time. We just described the process of *cultural evolution*, underlying the dynamics of social conventions. Cultural evolution derives from the repeated interactions between individuals of the population, everyone learning from and adapting to their interlocutor. These interactions can be between people of the same generation, designated as horizontal transmission, or different generations, what is called vertical transmission.

The concept of learning itself, from both a psychology or computer science point of view, has recently been studied as associated with intrinsic motivation, in the form of curiosity or novelty-seeking. Such intrinsically motivated behavior can allow learners to avoid learning situations that are too complex or too simple, and acquire skills and knowledge faster. In other words, learners can exhert a direct control on what and how they learn: we talk about *active learning*.

Do such mechanisms exist as well in the negotiation of social conventions, which are perceived as a form of learning? And do they have a significant impact on the associated dynamics? In this thesis, we will focus on a particular example of fast-changing social convention: the lexicon. We will insert active learning mechanisms in existing computational models of language emergence and lexicon negotiation, and show that they improve the global dynamics, in terms of both speed to reach an agreement and memory needed in the process.

1.2 Curiosity and intrinsic motivation

1.2.1 Intrinsic motivation

Childhood is filled with many challenges, one of the biggest being to learn an enormous quantity of skills, knowledge, and ways to interact with the world; all of that in a limited time and at an impressive pace. As an example, children typically learn an average of 8 new words a day before they turn 18, to reach the impressive total of roughly 60.000 (Bloom, 2002). Before that, they had to learn pronounciation and master their vocal tracts and all the muscular machinery needed to produce speech; as well as develop their auditory system and attention to perceive and select the words uttered by their peers. All these tasks are highly complex, and acquiring sufficient information in the world to efficiently learn how to solve them is not a trivial problem.

Young children naturally spend a lot of time playing and exploring their surroundings. They are switching from one activity to the other, and are subject to a constant stream of new information. They quickly become interested in something, but also get easily bored and switch to something else.

But how do children decide what new activities to pick? Their motivation in these examples is not external, but *intrinsic* (Berlyne, 1960; Ryan and Deci, 2000):

> Intrinsic motivation is defined as the doing of an activity for its inherent satisfaction rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures or rewards. (Ryan and Deci, 2000)

Spontaneously touching, grasping or throwing objects, babbling sounds, or later running around fall into the category of intrinsic motivation. Adults also can have intrinsically motivated behaviors: read a book, go for a hike, or play on their smartphone. Extrinsic motivation on the contrary would be linked to external pressures or rewards, as hunger, fear, or sleep.

Theories developped to define intrinsically motivating activities include the concept of *Flow* (Csikszentmihalyi, 1991): activities that are too difficult may trigger anxiety, and too simple ones boredom. In a learning context, new skills get acquired and gradually become easier. Other tasks on the contrary may have been too complex but are now within reach, thanks to the newly acquired skills. This process makes the optimal *Flow* zone gradually shift over time (see figure 1.2.1).

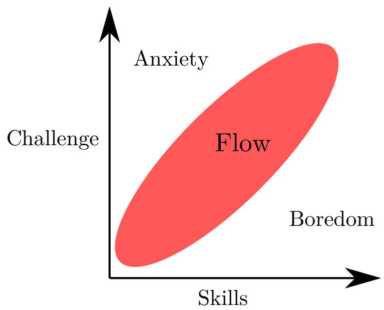


Figure 1.1: Simplified illustration of the *flow* (Csikszentmihalyi, 1991): given a certain level of skills, challenges can be too complex and cause anxiety, too simple and cause boredom. The *flow* is an intermediate zone, where learning can be more efficient.

This idea of an intermediate state is shared by other theories, based for example on optimal incongruity, where prefered situations are neither too certain nor too uncertain (Hunt, 1965). In Berlyne, 1965, it has been argued that many of the terms associated with intrinsic motivation (*incongruity*, *complexity*, *novelty*, ...) could be described as information theoretic variables. In practice, statistical properties of the stream of incoming information to the brain could be the source of an intrinsic reward system, based on the information carried by this input. It has also been argued to be an evolutionary advantage (Oudeyer and L. B. Smith, 2016): Among the many skills that can be learned by mere curiosity or play, some of them might be useful later, and will be readily available when needed. For example, lion cubs playfully fighting each other would reuse the same skills to either defend their territory or catch preys; human children try to imitate the sounds that they hear, which helps to later be able to pronounce the language of their parents.

Different functionals can be considered as intrinsic reward, and have been reviewed in Oudeyer and Kaplan, 2009. They include for example novelty predictivity (Barto, Singh, and Chentanez, 2004; Thrun, 1995), competence maximization (Oudeyer, Kaplan, and Hafner, 2007), or learning progress (Oudeyer, Kaplan, and Hafner, 2007; Schmidhuber, 1991).

1.2.2 Robots, algorithms and models

An alternative to measuring intrinsic rewards in the brain, which can be complex and invasive, is to consider robots, and more generally learning algorithms. Like babies, robots interact with their environment through a body whose actions through numerous actuators (motors or muscles) are sampled from a high-dimensional motor space. Typically, engineers program robots so that they can directly perform basic tasks [1] taking into account the mechanical properties of the body. Without those algorithms and parameters, the robot also has to learn how to interact with its environment. Luckily, in this situation, all internal states, functions, and rewards can be monitored, and the performance of different intrinsically motivated behaviors can be tested.

[1] e.g. walking, grasping objects, or responding in a certain way to some commands

Classic approaches to Machine Learning and Artificial Intelligence often include batch-processing enormous amounts of data, and with the development of computer superclusters neither memory usage nor computing time are even remotely comparable to the resources available to a child or a robot interacting with the real world[2].

[2] As an example, the AI algorithm Alphago who beat Go world champions (Silver et al., 2016), used the data of 160.000 recorded games, but only as a starting point: it simulated much more games, playing against itself or other AIs. It is impossible for a human brain to gather data from such a number of games, even in a lifetime. Despite this, world champions can still compete with AlphaGo and sometimes win.

How do humans solve this problem? Even tasks that we naturally consider really simple, because we execute them in our daily life without even thinking, are in reality very difficult. As an example, walking is harder than we imagine, and involves subtle balance and coordination between muscles. Robotic bodies are usually designed to make it possible for engineers to calculate efficient models and parameters for walking (stronger motors, lower center of gravity, shape of the legs, etc. – see figure 1.2 for an illustration). A child on the contrary learns in about a year how to stand up and walk, let alone discover its body and adapt to it, as it is at the same time quickly changing and growing.

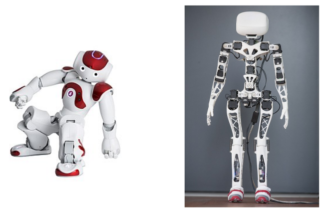


Figure 1.2: Right: The Nao robot (SoftBanks Robotics) is a commercial robot, whose legs and feet were designed to let it walk easily. Left: The Poppy humanoid robot was on the other hand designed to study human walk, and has body proportions and motor strengths resembling those of a child, but cannot walk yet without assistance (Lapeyre et al., 2014).

Behaviors inspired from the theories on *intrinsic motivation* can be implemented as algorithms and used in particular in robots: for example in the *Playground experiment* (Oudeyer and Kaplan, 2006), a robot is placed in a baby's playground and learns actively how to interact with its environment, by deciding which activities are more interesting than others, following a learning progress measure. There are two complementary sides to this kind of approach, called *developmental*: robotic experiments that let us better understand infant behavior by reproducing associated patterns (switching activities, developmental pathways, etc.), and adapting what is known in psychology to design more efficient algorithms for machine learning and AI in general, testing them on either robots or computational models.

Language acquisition has been one of the subject of such developmental experiments and models. For example in the *Playground experiment* (Oudeyer and Kaplan, 2006), introduced in the previous paragraph, the learning robot not only learns how to touch or grasp objects but also to vocalize to a peer, while not knowing at the beginning which of the available motor commands (muscle activations for humans) are associated to sound production. The experiment shows that active learning and intrinsic motivation can help make this distinction faster, along with a first refinement of vocal learning by imitation. Further studies showed that infant vocal development phases (Kuhl, 2004; Oller, 2000) can be observed in computational models of vocal learning in using intrinsic motivation (Howard and Messum, 2011; Moulin-Frier, Nguyen, and Oudeyer, 2014).

Figure 1.3: The Playground experiment (Oudeyer and Kaplan, 2006): a robot chooses its activities over time, according to its learning progress. Activity include grasping and touching objects, as well as communicating with a peer.

Lexicon learning as well can be impacted by active learning. As mentioned earlier, children build a lexicon of 60.000 words by the time they reach 18 (Bloom, 2002). It has been shown that they in fact can accumulate enough linguistic experience to do this: they hear enough occurences of words to have heard all distinct words at least once, if they all have the same probability of appearing in speech (Richard A. Blythe, A. D. M. Smith, and K. Smith, 2016; Richard A. Blythe, K. Smith, and A. D. M. Smith, 2010). However, if we consider the observed statistical distribution of words in speech, a powerlaw (Zipf, 1949), this result does not stand anymore. An alternative model shows that it is still possible to reach the final lexicon size when choosing actively between different learning situations, each associated to a different distribution of words (Hidaka, Torii, and Kachergis, 2017). This type of active behavior has been observed experimentally in adults (Kachergis, Yu, and Shiffrin, 2013).

1.3 *Computational models of language evolution*

Language is generally taught at school as a static set of rules and definitions, often including long lists of exceptions and irregularities. Those rules may however be subject to variations and deletion/creation of rules, and not because an authority like the *Académie française* decides it, but because language users spontaneously and unconsciously change their habits. New words and rules can be passed on to other fellow speakers who would on their turn propagate them and may change them further; or refuse to adopt them and stick to their own rules. Rules can replicate and propagate, but are also in competition against each other for at least one limited resource: language users. In other words, language is subject to an evolutionary process.

If biological evolution can be related to certain aspects of language (e.g. the shape of the larynx, or specific parts of the brain), it cannot account for the fast evolution of language: the usage of a new word can easily propagate across the globe in the matter of hours. This also reveals another aspect of language evolution that contradicts classic genetic inheritance of information: language can be transmitted directly from one person to the other, or *horizontally*, as opposed to *vertical* transmission from one generation to the other. Language can be described as subject to *cultural evolution*, and results from the interactions between individuals.

This conception of language has given birth to a variety of theoretical models, among which we distinguish *Iterated Learning* – focused on vertical transmission of language – , and *Language Games* – focused on the self-organization properties linked to horizontal transmission.

1.3.1 Existing models

Iterated Learning (Kirby, 2001; Kirby, Griffiths, and K. Smith, 2014; Kirby and Hurford, 2002) focuses on the evolution of language through vertical transmission, from generation to generation. Its exact definition is the following:

> Iterated learning: the process by which a behaviour arises in one individual through induction on the basis of observations of behaviour in another individual who acquired that behaviour in the same way.
> (Kirby, Griffiths, and K. Smith, 2014)

In other words, after learning a language through some example, a learner becomes a teacher by producing a few examples of their own interpretation of the language that will be presented to a new individual. The language samples presented to the first learner are usually random in the space of possible signals. Depending on the number of samples that are presented to each learner, also called *bottleneck*, there is a pressure on the language for compressibility. Computer simulations of the models show that this pressure can account for the emergence of compositionality in language (Kirby, 2001; Kirby and Hurford, 2002).

The term of language games was introduced by Wittgenstein, 1953 as a description for many of our daily interactions that may either use language or constitute a communicative substrate for the emergence of language. Within computational models of language evolution, it refers to a class of multi-agent models (Steels, 1995) in which language is seen as a complex adaptive system (Steels, 2000). They focus on the self-organizational properties resulting from the horizontal transmission of language, and interestingly show that some features of language *can* emerge spontaneously within a population of agents, without a centralized control or a partially innate language.

One of the simplest version is called the *Naming Game*: agents gradually agree on a lexicon, a mapping between a set of signals and meanings (Loreto, Baronchelli, et al., 2011; Steels, 1995; Wellens, 2012). This model being at the core of this thesis, chapter 2 will be dedicated to its detailed description.

A great taxonomyy of variants of the Language Games exist, showing that a linguistic system can arise for colors (Bleys et al., 2009; Puglisi, Baronchelli, and Loreto, 2008; Steels, Belpaeme, et al., 2005), for spatial representations (Spranger, 2012) or for grammatical features (Beuls and Steels, 2013; Cuskley et al., 2017; Van Trijp, 2012).

1.3.2 *Comparisons to real data and behavior*

Those models were built to try to describe human behavior, but can we compare them to real data? There are two possible ways: either compare results of the models to existing databases about natural languages, or conduct user experiments to directly monitor the behavior of participants placed in the context of one of the models.

To compare the models to existing databases, we can reproduce similar data through many simulations of the models – usually Language Games – and compare their statistical properties. This has been done for example for vowel systems (De Boer, 2001; Oudeyer, 2006) using their number and distribution in vowel space in natural languages, color naming systems (Baronchelli, Gong, et al., 2010; Puglisi, Baronchelli, and Loreto, 2008) using a large scale survey of statistical properties of unwritten languages (Kay et al., 2009), or the emergence of creole languages (Tria, V. D. P. Servedio, et al., 2015) using census data from the United States.

Experimental semiotics (Galantucci and Garrod, 2012) refers to user experiments where participants develop a new artificial communication system. Such experiments can be either conducted by inviting participants to the lab, or online as a web application, potentially submitted to a crowdsourcing platform massively recruiting participants.

The Iterated Learning paradigm gave birth to many user experiments, showing that while passing through chains of participants languages do evolve to be more easily learnable, through the emergence of compositionality. The first experiment involved artificial languages about colored shapes moving along certain trajectories (Kirby, Cornish, and K. Smith, 2008), compositionality could arise to refer to those three features (shape, color, motion). In other experiments, different types of signals[3] were used, for example sounds made with a slide whistle (Verhoef, Kirby, and De Boer, 2014) or graphical shapes with an inner structure (Cuskley, 2018).

[3] compared to typed text for the original experiment

Language Games and similar communication tasks using horizontal transmission were also adapted to user experiments. For example the structure of the social network regulating interactions between the participants can influence spreading dynamics (Centola and Baronchelli, 2015), pictionary-like experiments result in simpler and more abstract drawings (Garrod et al., 2007), or the emergence of compositionality with only horizontal transmission (Raviv, Meyer, and Lev-Ari, 2019). Some experiments have also been conducted online, presented as actual games (Cuskley, 2018; Morin, Winters, T.

Müller, et al., 2018).

1.3.3 Active learning in Language Games

A first approach to the introduction of active learning in Language Games can be found in Steels, 2004 as the *autotelic principle*. In the task they have to accomplish, agents can distinguish different *challenges*. For example when naming objects that have a shape and a color, refering only to the object class (e.g. a table) is easier than refering to both the class and the color. This idea was implemented and showed that agents can reach an agreement faster, if they structure their interactions by selecting the challenges according to a policy making them stay in the *flow* zone (Cornudella and Poibeau, 2015; Steels and Wellens, 2007). This also works in another setting: labelling the space of colors. In practice, colors can be referred to either in an holistic way, by adding modifiers to known color terms, or mixing several known colors; those levels being the considered set of challenges (Cornudella Gaya, 2017). This approach considers that challenges and their difficulty levels are known beforehand by all agents.

Another approach was introduced by Oudeyer and Delaunay, 2008, which relies on less knowledge being pre-shared by agents: active topic choice. When having to name several objects, agreement can be reached faster if agents first settle for the labels of some objects, and then move to naming other ones. This approach will be the core element of this thesis.

1.4 This thesis

1.4.1 Overview and structure

This thesis uses a computational model, the Naming Game, and a specific approach to the introduction of active learning behavior: active topic choice (Oudeyer and Delaunay, 2008). We focus on two properties of these policies: agreement can be reached faster, and less memory is needed.

Chapter 2 presents the model, and the many variants that have been studied, along with relevant measures and properties of the model. In particular, the time to reach a global agreement and the maximum memory needed by agents evolve as powerlaws of the size of the population. To stay cognitively plausible, memory should stay low, and not grow as a power of the population size.

Chapter 3 introduces the notion of active topic choice, and exhibits a few policies under hard memory constraints: homonyms and synonyms are forbidden. Information Gain maximization and Chunks strategies only rely on the current size of the lexicon of the agent. Success Threshold and Minimal Counts rely on a count of the past successes and failures of communication, respectively in a relative way or an absolute way. We show that the last two can converge significantly faster than a random topic choice policy.

Chapter 4 reconsiders homonymy and synonymy: the aim is to design new strategies that will themselves control memory usage. A new measure is introduced, the LAPS measure, or Local Approximated Probability of Success. It represents an agent-level estimation of the level of agreement with the whole population. A strategy maximizing this measure is presented, and shows remarquable properties in terms of convergence time and memory usage. The LAPSmax strategy however does not use direct maximization of the LAPS measure, but a heuristic: it would be too difficult to compute otherwise. A simpler measure, close to LAPS and called coherence, leads to another strategy with direct maximization. This policy converges even faster than LAPSmax, but uses a bit more memory.

Chapter 5 introduces statistical estimations of some of the measures. In particular, we calculate a statistical lower bound to the convergence time. We use these expressions to define six performance measures, characterizing convergence time and memory usage. We compare the strategies of previous chapters, and show that the Coherence strategy is near optimal.

Chapter 6 introduces a user experiment that was conducted to monitor the behavior of real participants in the context of a Naming Game, as an online web application. Because we are only interested in the decisions of the agents, we can simulate the rest of the population and easily conduct the experiment with individual participants. Observed behaviors show less exploration than a random topic choice policy.

1.4.2 Contributions

Naming Game classification: We introduced a new classification of existing Naming Games, explicitly isolating the agent side from the population.

Implementation of the classification: The classification was implemented in the form of a modular open-source Python library, allowing to easily extend the work presented in this thesis to other parameters or hypothesis, or study the Naming Game under other aspects. Results from earlier work (mainly Baronchelli, 2006) were reproduced using this library. Databases of most of the experiments used in this thesis will be provided online[4] to allow curious readers to further analyze them. It represents the equivalent of several years of CPU time.

[4] Github: wschuell/notebooks_thesis

Designing efficient strategies: Several classes of strategies were introduced, each of them having advantages and drawbacks in terms of memory usage and convergence speed. Within correct parameter ranges, they have better properties than a random topic choice policy.

Relevant measure: We defined a relevant local measure as a functional to optimize for agents, the LAPS measure (Local Approximated Probability of Success).

Theoretical analysis: Statistical lower bounds were exhibited, associated to performance measures to classify the different strategies.

Design of user experiment: To our knowledge, this is the first time that a Language Game user experiment has been conducted with simulated agents, to study not the global properties of the language or the agreement process but only the local behavior.

User experiment results: We have shown that participants were refraining from exploring new meanings, as compared to a random topic choice behavior.

User experiment framework: The user experiment was implemented as a web application. It is dockerized[5], and deployable on a new server in a few minutes. It can be modified and reused for other configurations of the Naming Game, relying directly on our modular Naming Game library.

[5] Docker containers are a way to package software to ensure that it will run in the same way on many different computer systems.

Simulation management framework: We have implemented an open-source Python framework for efficiently managing experimental campaigns based on computational models: management of parameters, scalable on a computing cluster, storage of results, etc.

1.4.3 *Publications*

- William Schueller and Pierre-Yves Oudeyer (2015). "Active learning strategies and active control of complexity growth in naming games". In: *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE. DOI: 10.1109/devlrn.2015.7346144
- William Schueller and Pierre-Yves Oudeyer (2016). "Active Control of Complexity Growth in Naming Games: Hearer's Choice". In: *EVOLANG 2016*. Proceedings for the 11th International Conference on the Evolution of Language (EvoLang XI). New Orleans, United States
- William Schueller, Vittorio Loreto, and Pierre-Yves Oudeyer (2018). "Complexity Reduction in the Negotiation of New Lexical Conventions". In: *40th Annual Conference of the Cognitive Science Society (CogSci 2018)*. Madison, WI, United States

1.5 *Short summary*

Due to the interdisciplinary nature of this thesis, the reader may be interested in some parts more than others, depending on their background. Here are a few indications of how to interpret this work, in the form of a short summary for several disciplines.

1.5.1 For computer scientists

This thesis revolves around a computational problem: decrease memory usage and computation steps of a distributed algorithm. The associated multi-agent model and the problem itself are detailed in chapter 2. A first approach with imposed hard memory constraints is proposed, with a few algorithms, in chapter 3. Two other strategies are presented in chapter 4, based on a local[6] measure of estimated success of communication. Both of those strategies perform well in terms of memory and computation time, compared to a theoretical optimal strategy, which is developped in chapter 5. One strategy requires a bit more memory, while the other requires a bit more computation: depending on one's needs, one or the other could be used. Eventually, chapter 6 presents a user experiment gathering data on the spontaneous behavior of human participants faced with the same problem.

[6] at the agent level

An important remark is that compared to other AI models of learning agents, a lot of agents have to be simulated at the same time to potentially optimize policies or other parameters. Scaling memory and computation time to large populations quickly becomes an issue.

1.5.2 For physicists and complexity scientists

This thesis studies modifications to an existing multi-agent model. This model was previously studied and described with a statistical mechanics approach (Baronchelli, 2006), leading to a characterization of its asymptotical behavior with increasing population size. The variant of the model leaves room for optimization in terms of memory usage (by the agents) and number of interactions necessary to reach the stable state, depending on agent behavior. After a description and taxonomy of the model in chapter 2, a first set of strategies under memory constraints are presented in chapter 3. A second series of strategies are introduced in chapter 4, based on the local estimation of a global functional (i.e. population level) using information sampling. The latter show near-optimal dynamics, by comparison with a statistical lower bound to the time needed to reach the final stable state, calculated in chapter 5. Eventually, in chapter 6, an online user experiment allowed us to collect data on human behavior when interacting as in the model. Participants were playing with simulated agents, as we were interested only in the microscopic behavioral patterns of agents; to be able to compare them to the strategies of the previous chapters.

Open-source code (see last paragraph of this section) includes a framework to manage experimental campaigns of simulations of the model, that can be adapted to other models written in Python. This allows the user to for example run their code transparently on a computing cluster, without the hassle of debugging the experiment management part or confusing it for errors in the main code, updating the code on the cluster, or collecting data and keeping track of parameters and configurations.

An important remark is that the local models of the agents are a bit complex, as they use a local memory and heuristics that can be a bit elaborate: it is less easily scalable in terms of computation time than usual complex systems and physics models.

1.5.3 *For cognitive scientists*

This thesis introduces active learning mechanisms, inspired the field of developmental psychology, in a multi-agent model describing the collective construction of a language. The model is described in chapter 2, and several strategies are introduced, either under memory constraints (chapter 3) or maximizing an estimated probability of having a successful communication (chapter 4). Limiting memory usage is necessary for the algorithms to be cognitively plausible. Using a theoretical statistical lower bound (chapter 5), we show that strategies of chapter 4 are comparable to optimal. Eventually, those strategies are compared to human behavior through an online user experiment (chapter 6), showing that humans indeed exhert an active control on their learning situation when incarnating an agent in the model.

The user experiment is available as open-source code (see last paragraph of this section), and can be deployed easily in only a fews minutes on a new server. Modifications to study variants of the model or the experimental setup can be integrated easily.

1.5.4 *Complement for everyone*

This thesis gave birth to open-source code, written mainly in Python:

NamingGamesAL library: Modular library for simulation of the model (Naming Games with Active Learning). Modularity gives the possibility to quickly develop a variant of the model, by making easy to isolate and modify small parts of it in the code.

Experiment manager library: Making easier experimental campaigns, through optional submission to computing clusters in an adaptive way, storing results, and managing parameters efficiently.

User experiment : Dockerized and easily deployable/stress-testable/readily analysable version of the experiment. It relies on the NamingGamesAL library, and can be directly used to modify the characteristics of the experiment.

Unless precised explicitly, the data of all curves and quantitative figures presented in this thesis have been produced using those libraries.


Evolang Conference, New Orleans, LA. April 2016

2
The Naming Game

Contents

2.1 *General description of the model*

The Naming Game (Baronchelli, Felici, et al., 2006; Steels, 1995) is a computational framework aiming at describing dynamics of lexicon self-organization in a population of interacting agents. At each interaction, two agents are randomly selected from the population and designated as speaker and hearer. They try to communicate about a predefined and finite set of meanings, using words from a predefined and finite set. Only one meaning is selected for the interaction, and is called the topic. According to the outcome — success or failure of the communication — they align/update their own lexicon. For an overview, see figure 2.1. A more concrete and minimalist definition, taken from (Wellens, 2012), is the following:

1. Each interaction happens between only two participants (speaker and hearer)
2. In each interaction, a single word is used to refer to a single meaning (the topic)
3. Meanings (and words) do not have internal complexity or features
4. At the end of the interaction, both participants know the intended topic
5. In each agent's vocabulary, several competing word-meaning associations can coexist (synonymy and homonymy)

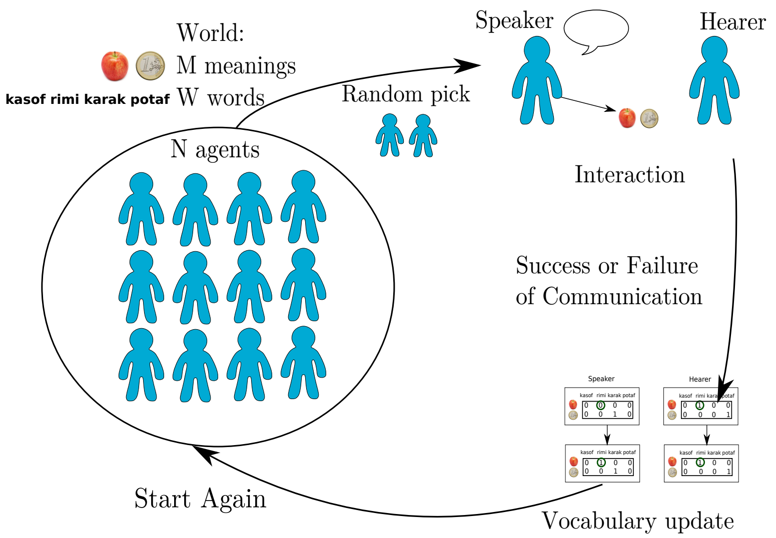


Figure 2.1: Overview of a NG model: Convergence towards a global pattern (shared lexicon) through repeated local interactions. For a more detailed descriptions of the different parts and alternatives, see figure 2.14.

This definition still lets room for A LOT OF POSSIBLE INTERPRETATIONS AND IMPLEMENTATIONS, and in fact many of them have been studied: different types of vocabulary update policies (Wellens, 2012), picking interacting agents in a non-uniform way (Dall'Asta et al., 2006), having non-uniform word selection policies (Baronchelli, Dall'Asta, et al., 2005), or having agents replaced/introduced in the course of the simulation (Steels and Kaplan, 1998; Vogt and Coumans, 2003). We will in the following sections describe in more details those aspects, and explain what we will consider as the standard implementation, or minimal NG. An overview of the classification that we used can be seen in the last section of this chapter, figure 2.14. In case of doubt, the code used to run all simulations presented in this thesis is available as open-source software at github.com/flowersteam/naminggamesal.

AGENTS in general will be noted $\mathcal{A}$, or $\mathcal{S}$ and $\mathcal{H}$ when they will be identified as respectively speaker or hearer. We will identify an agent by its lexicon, hence both will share the same notation. We will note the population $\mathfrak{P}$ composed of N agents:

$$\mathfrak{P} = (\mathcal{A}_i)_{1 \leq i \leq N} \tag{2.1}$$

N will generally be equal to 100 throughout this thesis, or span from 10 to 1000 when studying the influence of population size. This choice will be explained in section 2.1.2.

2.1.1 *Meaning and word spaces*

Meanings and words are typically discrete elements from a finite set. As mentioned in the definition above, they do not have internal complexity, they cannot be referred to as a group sharing a specific feature. As a consequence, lexicons used in the simulations are holistic, as combinatorial languages would have by definition internal complexity in the space of words.

Words will be noted w_i or w. In most cases (and in what we consider the standard Naming Game), the word space $\mathfrak{W}$ is a finite set of size W, without any structure:

$$\mathfrak{W} \equiv (w_i)_{1 \leq i \leq W} \tag{2.2}$$

We distinguish two cases: $W = M$ and $W \gg N \cdot M$ where M is the number of meanings, and N the number of agents in the population. The latter can be considered as a *pseudo-infinite* set of words. In this case, two independent *inventions* of a word (picking one that is not used yet) cannot result in picking the same word. In other words, there is no possible homonymy. The choice of values for W will be discussed in section 2.1.2.

Meanings will be noted m_i or m. In most cases (and in what we consider the standard Naming Game), the meaning space $\mathfrak{M}$ is a finite set of size M, without any structure[1]:

$$\mathfrak{M} \equiv (m_i)_{1 \leq i \leq M} \tag{2.3}$$

M will generally be equal to 100 throughout this thesis. This choice will be explained in section 2.1.2.

[1] In some cases, the meaning space $\mathfrak{M}$ can be given a structure. Individual meanings are nodes in a graph, or they can show a different relative *utility*: agents are biased when picking one of them as a topic.

2.1.2 *Values for N, M and W*

The behavior of the system depends on the three variables N (population size), M (number of meanings) and W (number of words). The nature of the problem can differ if the relation between those three variables differ. For example, if $N \ll M$, few agents are faced with the problem of exploring a huge space of meanings, and could divide efficiently the task of labelling all of them; whereas if $N \gg M$, many conflicting conventions can be invented at the same time and the main issue shifts to resolving those conflicts. In practice, we will set $M = 100$, and use values of N ranging from 10 to 1000 in order to cover both cases, while including the intermediary case $N = M$.

In previous analytical work (Baronchelli, 2006), $M = 1$. They justify this choice by saying that if the word space is pseudo-infinite, the same word cannot be invented twice for two distinct meanings. Without homonymy, the dynamics of the Naming Game for $M > 1$ meanings is the same as for 1 meaning, it is only slower by a factor M. This relies on another strong assumption: topics are always picked uniformly from the meaning space $\mathfrak{M}$. When using an active topic choice policy (see paragraph 2.3.3 and chapter 3), that is not true. As one of the main contribution of this thesis, it even relies on $M > 1$.

To allow convergence towards a complete lexicon without conflicting synonyms or homonyms, there should be at least as many words as meanings, i.e. $W \geq M$[2]. We will in this thesis mostly consider the case $W = \infty$. In practice, that means $W \propto M \cdot N$, i.e. the maximum number of words that can be invented in a simulation.

[2] See next paragraph for a detailed definition of lexicons.

One could argue that we can have $W \approx M$ when considering language emergence, or the bootstrapping of a first lexicon: the set of available signals may be restrained (a few vocalizations for example). It would be only afterwards, driven by a need of more signals to describe an expanding space of meanings, that W would expand as well. In natural languages, the space of words or signals is combinatorial, and so is unbounded. However, we tend to choose/invent words in a subset of possible words: for every language, there is a heavy bias towards certain letter combinations over others. Also, there is a natural drive towards selecting short words, as longer ones would be more difficult to remember and would need more time to pronounce. The restrained space of eligible new words may however still be greater than the potential set of new meanings considered, but we nevertheless observe homonymy in natural languages.

In the Naming Game, the possibility to restrain the set of words could help: there is a possibility that two independent inventions would use the same word. But on the other hand, this also increases consequently the probability of having conflictual conventions. We will study the impact of restraining the size W of the set of words through the settings $M = W$ and $M = 2W$.

2.1.3 *Vocabulary representation*

VOCABULARIES, or lexicons, are in the model a set of associations between meanings and words. In the context of finite sets of words $\mathfrak{W}$ and meanings $\mathfrak{M}$, vocabularies can be represented as associations matrices, where each row corresponds to a meaning, and each column to a word. This representation has been extensively used in related work (Ke et al., 2002; Oliphant and Batali, 1997; Steels and Kaplan, 1998). Two parts of the lexicon are distinguished: the coding or production part, which maps a meaning to a set of words weighted by probabilities of usage, and a decoding or interpretation part, mapping a word to a set of meanings that can be interpretated from this word, also weighted by probabilities. In practice, the real values — also called *scores* — present in the matrices are not directly the values of the probability weights: Either they are normalized (by row for coding and by column for decoding), or the probability weights are distributed uniformly among the coordinates having the maximum score (again, per row for coding and per column for decoding). In practice, the latter is used.

We represent the vocabulary of an agent $\mathcal{A}$ as a matrix of size $M \times W$, with values in $[0;1]$ for each word-meaning association used by the agent. We will here identify the agent to its matrix, and call it $\mathcal{A}$. Each agent starts with an empty vocabulary, a matrix filled with zeros. The coding matrix $\mathcal{A}^c$ and decoding matrix $\mathcal{A}^d$ are derived from $\mathcal{A}$ by normalizing respectively over rows and columns:

$$\mathcal{A}^c_{mw} = \frac{\mathcal{A}_{mw}}{\sum\limits_{w'} \mathcal{A}_{mw'}} \qquad \mathcal{A}^d_{mw} = \frac{\mathcal{A}_{mw}}{\sum\limits_{m'} \mathcal{A}_{m'w}} \tag{2.4}$$

Normalization factors are used only if $\mathcal{A}_{mw} \neq 0$. In practice, when coding a meaning m, a word w_i is sampled using the distribution $(\mathcal{A}^c_{mw})_{w \in W}$. When decoding a word w, a meaning m_j is interpreted, sampled from the distribution $\left(\mathcal{A}^d_{mw}\right)_{m \in M}$. In our case, these distributions are uniform either on the set of words associated to m for coding, or on the set of meanings associated to w for decoding. Those two sets change over time, during the vocabulary update.

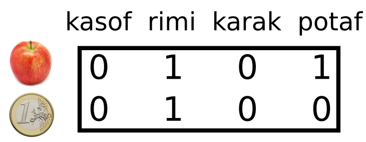


Figure 2.2: Illustration of a lexicon represented as an association matrix between the meanings m_i and the words w_j. There are two conflicts: *rimi*/*potaf* are synonyms, they both refer to the apple; and *rimi* is subject to homonymy, as it refers to both the apple and the coin. In practice, two matrices are used: one for coding meanings into words, and one for decoding words as meanings. They can be deduced from the presented matrix by normalizing respectively over rows and columns. In some versions of the model, values can be real values between 0 and 1. In this work we will mainly consider binary values, this matter being discussed in paragraph 2.3.1.

2.1.4 *Scenario*

Each interaction involves two agents: Speaker ($\mathcal{S}$) and Hearer ($\mathcal{H}$). They are beforehand picked from the population, usually randomly[3]. $\mathcal{S}$ and $\mathcal{H}$ follow a given scenario in five steps:

[3] The alternative is to define an underlying social network, and pick them so that one can only interact with its direct neighbors in the network. See paragraph 2.4.1

1. $\mathcal{S}$ chooses a topic m_S
2. $\mathcal{S}$ checks its vocabulary to find or invent a word w associated to m_S
3. $\mathcal{S}$ utters the word w
4. $\mathcal{H}$ guesses a meaning m_H from w using its vocabulary.
5. $\mathcal{S}$ indicates the intended meaning m_S (by pointing at it for example)

After these steps, agents can have feedback whether the communication was successful or not. A realistic analogy could be that both can see if m_S and m_H match if $\mathcal{H}$ is startled or not, or if $\mathcal{H}$ indicates back m_H.

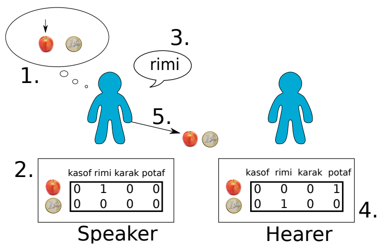


Figure 2.3: Interaction scenario of the standard NG. Beforehand, two individuals have been randomly selected among a population, and designated as speaker ($\mathcal{S}$) and hearer ($\mathcal{H}$). See main text for the detailed description of the five steps.

The main contributions of this thesis are made by intervening on the first step, and letting $\mathcal{S}$ choose actively the topic instead of randomly. This will be explained in greater details in paragraph 2.3.3 and then in chapter 3.

An alternative scenario exists, called *Hearer's choice* where $\mathcal{H}$ is picking the topic and indicating it to $\mathcal{S}$. The rest of the interaction is pretty much the same[4] By contrast, the standard scenario is also referred to as *Speaker's choice*. They are both compared in paragraph 3.4.3.

[4] Precision: $\mathcal{H}$ still guesses a meaning m_H from w, and can directly compare it with m_S.

2.2 *Measures and general dynamics*

2.2.1 *Notion of convergence*

The self-organization process happening while simulating the Naming Game has complex dynamics, and goes through various states before reaching global consensus. We talk about those dynamics as a convergence process, towards a state where all agents share the exact same lexicon, with exactly one word for each meaning without synonymy and homonymy. Such a state is stable[5], lexicons will not change anymore whatever are the modalities of the interaction – which agent is the speaker, which is the hearer, and which meanings and words are used. Convergence and stability of the Naming Game has been proved analytically (Vylder and Tuyls, 2006).

[5] provided there is no introduction of new agents

An important remark at this point is that there exists a lot of possible fully-converged state[6], and it is precisely this that makes it hard for agents to solve the Naming Game and reach one same converged state.

[6] Exactly $\frac{M! \cdot W!}{(W-M)!}$, see section 3.2.3 for more details.

In some variations of the NG, full convergence cannot be reached at reasonable time scales [7], or even not at all. However, a stable or pseudo-stable[8] state can still be reached and characterized[9], we then talk about partial convergence.

[7] Comparable to other dynamics of the system, mainly of spreading most of the competing conventions, before starting to discard some of them

[8] Compared to the said time scales

[9] Using mainly the *TCS* measure introduced in the next paragraph

In this thesis, we do not focus on whether the model converges or not, but on the speed and complexity properties of the dynamics before convergence. Measures for each of those aspects, used to describe the system while in this intermediate state, can be found in previous work (Loreto, Baronchelli, et al., 2011). We distinguish local measures – accessible to each agent – from global measures, computed on the whole population.

2.2.2 *Theoretical communicative success*

The Theoretical Communicative Success, or *TCS*[10], is a measure of distance to a fully converged state. First, for each meaning, we can consider the probability of having a successful communication when using this meaning as a topic, given a state of the population. The *TCS* is the average of those probabilities, over all possible meanings. In the case of Random Topic Choice, this measure coincides with the general probability of having a successful interaction. By definition, it is a global measure, not accessible to individual agents. To retrieve its value, we can either estimate it using a snapshot of the population and a Monte Carlo method with random topic choice, or compute it. To detail the exact computation formula, we

[10] A table of symbols, acronyms and abbreviations can be found at the last page of this manuscript.

need to first define the probability of success between two given vocabularies of agents $\mathcal{A}$ and $\mathcal{B}$. As detailed in the previous section, a vocabulary has two components: a coding part, used to find words associated to a meaning, and a decoding part, used to find meanings associated to a word. For vocabulary A, we would then have the two matrices $\mathcal{A}^c$ and $\mathcal{A}^d$. If $\mathcal{A}$ is the speaker and $\mathcal{B}$ the hearer, $\mathcal{A}$ is coding and $\mathcal{B}$ decoding, hence the formula of the probability of success in this case, averaged over all possible meanings:

$$TCS_S(\mathcal{A},\mathcal{B}) = \frac{1}{M}\sum_m\sum_w \mathcal{A}^c_{m,w}\cdot\mathcal{B}^d_{m,w} \tag{2.5}$$

If A would be hearer, we have by symmetry:

$$TCS_H(\mathcal{A},\mathcal{B}) = TCS_S(\mathcal{B},\mathcal{A}) = \frac{1}{M}\sum_m\sum_w \mathcal{B}^c_{m,w}\cdot\mathcal{A}^d_{m,w} \tag{2.6}$$

Because before an interaction we do not necessarily know which agent will be the speaker and which will be the hearer, the two situations ($\mathcal{A}$ speaker and $\mathcal{B}$ hearer / $\mathcal{B}$ speaker and $\mathcal{A}$ hearer) are to be considered as equiprobable. The final value $TCS(A,B)$ is the mean of those two values:

$$TCS(\mathcal{A},\mathcal{B}) = \frac{TCS_S(\mathcal{A},\mathcal{B}) + TCS_H(\mathcal{A},\mathcal{B})}{2} \tag{2.7}$$

To scale up to population level, one can compute an average vocabulary for the whole population $\mathcal{V}(\mathfrak{P})$, and then the probability of success for an interaction between this lexicon and itself. For a large enough population, this value is indeed a good approximation of the probability of success. $\mathcal{V}(\mathfrak{P}) = \langle\mathcal{A}\rangle_{\mathcal{A}\in\mathfrak{P}}$ is an element-wise average of the lexicon matrices of all agents. This measure taken over time, as in previous work, will be noted $S(t)$.

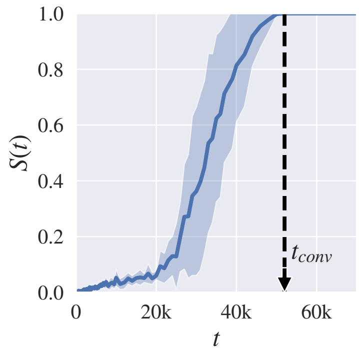


Figure 2.4: Typical evolution of the probability of success. M=1, $N = 1000$, $W = \infty$, averaged over 8 trials.

$$S(t) = TCS(\mathfrak{P}) = TCS(\mathcal{V}(\mathfrak{P}),\mathcal{V}(\mathfrak{P})) = TCS(\mathcal{A},\mathcal{B})_{\mathcal{A},\mathcal{B}\in\mathfrak{P}} \tag{2.8}$$

In the standard Naming Game, this value stays close to 0 for a time and then abruptly goes from 0 to 1 after a certain number of interactions, as seen on figure 2.4 (Baronchelli, Felici, et al., 2006). This transition happens when agents shift between a first phase of inventing conventions to a phase of discarding conflicting ones. These phases can be retrieved in other models of convention propagation, and were labelled in Fagyal et al., 2010: Innovation (inventing conventions), Selection/Propagation (the abrupt shift), and Fixation (discarding competing conventions). The typical evolution of the TCS value can be seen on figure 2.4.

2.2.3 *Agent-level local complexity*

For each agent, we can define a local complexity measure, by counting the number of distinct associations present in the vocabulary. In our case, this is exactly the sum of all elements of the matrix A. At the beginning of a simulation, while the vocabulary is empty, this measure equals 0. At the end, its value is the number of meanings M. When using random topic choice, there is a fast growth to a maximum, before a slow decrease to the final value M (can be seen in fig.2.4). This local complexity measure, when averaged over the whole population, is proportional to the measure N_w used in previous analytical work (e.g. Baronchelli, Felici, et al., 2006): $\langle LC(\mathcal{A})\rangle_{\mathcal{A}\in\mathfrak{P}} = \frac{N_w}{N}$.

This measure will be our measure of memory, as it it close to a minimal memory representation of the lexicon (e.g. as a sparse matrix or a list of word-meaning associations). Because if this, we expect it to remain low in a cognitively plausible situation, which is not the case in the standard dynamics. Throughout this thesis, we will focus on this very issue.

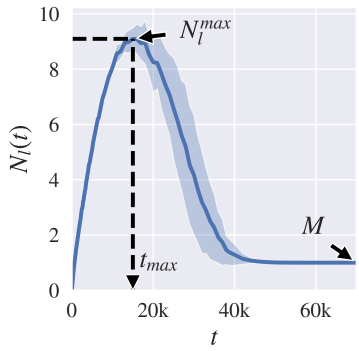


Figure 2.5: Typical evolution of the number of words per agent, or local complexity. M=1, N = 1000, $W = \infty$, averaged over 8 trials.

2.2.4 *Population-level global complexity*

We can extend this notion of complexity to the whole population, by counting conventions that are present at least once in one agent's vocabulary. Another way to see it is as the local complexity measure of the average vocabulary of the whole population, $\mathcal{V}(\mathfrak{P})$. This was already introduced in previous work (e.g. Baronchelli, Felici, et al., 2006) as the measure N_d. To sum up:

$$N_d = LC\left(\mathcal{V}(\mathfrak{P})\right) \tag{2.9}$$

Typically, N_d grows extremely quickly to a maximum before slowly decreasing to the stable value M. Its maximum is reached when all agents have invented conventions for each meaning, and is equal to $\frac{N\cdot M}{2}$ [11]. For more theoretical considerations about this measure, see chapter 5.

[11] This value is the count of the events *It is the first time the selected speaker talks about this meaning*. For a given agent, the probability to first interact about a meaning as speaker is 0.5, so if meanings are chosen randomly, there are $\frac{N}{2}$ inventions per meaning.

2.2.5 *Other measures*

A few of other measures can be derived from what we have seen so far:

Convergence time t_{conv}: Time, in number of interactions, to reach $TCS = 1$.

Maximum global complexity time t_d: Time, in number of interactions, to reach $N_d = N_d^{max}$.

Maximum local complexity time t_{max}: Time, in number of interactions, to reach $N_l = N_l^{max}$.

Observed Communicative success $S_O(t)$: Not to be mistaken for $S(t)$, they differ in the case of a biased topic choice.

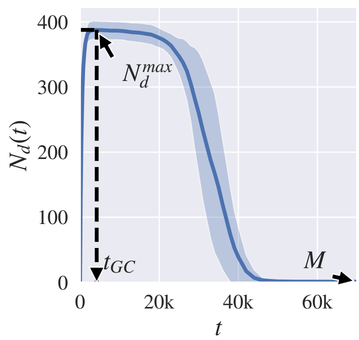


Figure 2.6: Typical evolution of the number of conventions present in the population, or global complexity N_d. $M = 1$, $N = 1000$, $W = \infty$, averaged over 8 trials.

Other ones will be used later in this thesis, but we can nevertheless mention a few of them here:

Number of inventions N_{inv}: Number of individual inventions. In the standard NG, it is equal to N_d^{max}.

Missing information $i(\mu, M, W)$**:** Quantity of information needed for an agent to complete a partially filled lexicon with μ associations, defined and used in chapter 3.

Entropy $E(t)$**:** Extension of the precedent measure including homonymy and synonymy, defined in chapter 4.

Success ratio $S_R(m,t)$**:** Local proportion of successes in the past interactions with meaning m, defined and used in chapter 3.

Success count $S_C(m,t)$**:** Local number of successes in the past interactions with meaning m, defined and used in chapter 3.

LAPS measure $L(t)$**:** Local Approximated Probability of Success, defined in chapter 4.

Coherence measure $C(m,t)$**:** A local measure of coherence of perceived word usage for meaning m in the last interactions, defined in chapter 4.

2.2.6 *Scaling*

When comparing the dynamics of several variations of a system depending on a set of parameters (here: N,M and W), it is important not to draw direct conclusions from a single example, but to explore different ranges of parameter values. Given a certain set of values for each parameter, it is yet difficult to even represent the data, let alone the combinatorial number of simulations that have to be run. A solution to this problem is to search for *laws* describing the relationship between the evolution of the system and its parameters. In many self-organizing systems, the dependence on each parameter is separated from the other parameters, and can often be a *powerlaw*. For example, many natural phenomenon either show power law patterns: the population of cities, the distribution of words in speech, earthquake events, or even many biological features can be described by powerlaws (Newman, 2005; West, 1997; Zipf, 1949).

In the Naming Game in particular, it has been shown that some measures do scale as powerlaws (Baronchelli, 2006). Values can be observed empirically, or in some cases approximated analytically. For the minimal Naming Game, we have the following scaling relations, some of which can be observed on figure 2.7:

- $t_{max} \propto N^{\alpha}$ with $\alpha \approx 1.5$
- $N_l^{max} \propto N^{\beta}$ with $\beta \approx 0.5$
- $t_{conv} \propto N^{\alpha}$ with $\alpha \approx 1.5$. However, we can see empirically[12] that this explonent is only well approached after $N > 10^4$, it is smaller otherwise. It necessarily reaches this value, as $t_{conv} > t_{max}$.

[12] Data not shown, can be found in Baronchelli, 2006. Further details can be found in chapter 5.

More detailed information and some analytical arguments can be found in chapter 5.

These values reflect a *computational cost* of the Naming Game, in terms of time of execution or memory needed. The main objective of this thesis is to study how they can be reduced. In an ideal case, the memory needed is of the order of magnitude of the final size of the lexicon $N_l^{max} = O(M)$, i.e. $\beta \approx 0$; and the time needed to converge stays the same for each agent, i.e. $\alpha \approx 1$.

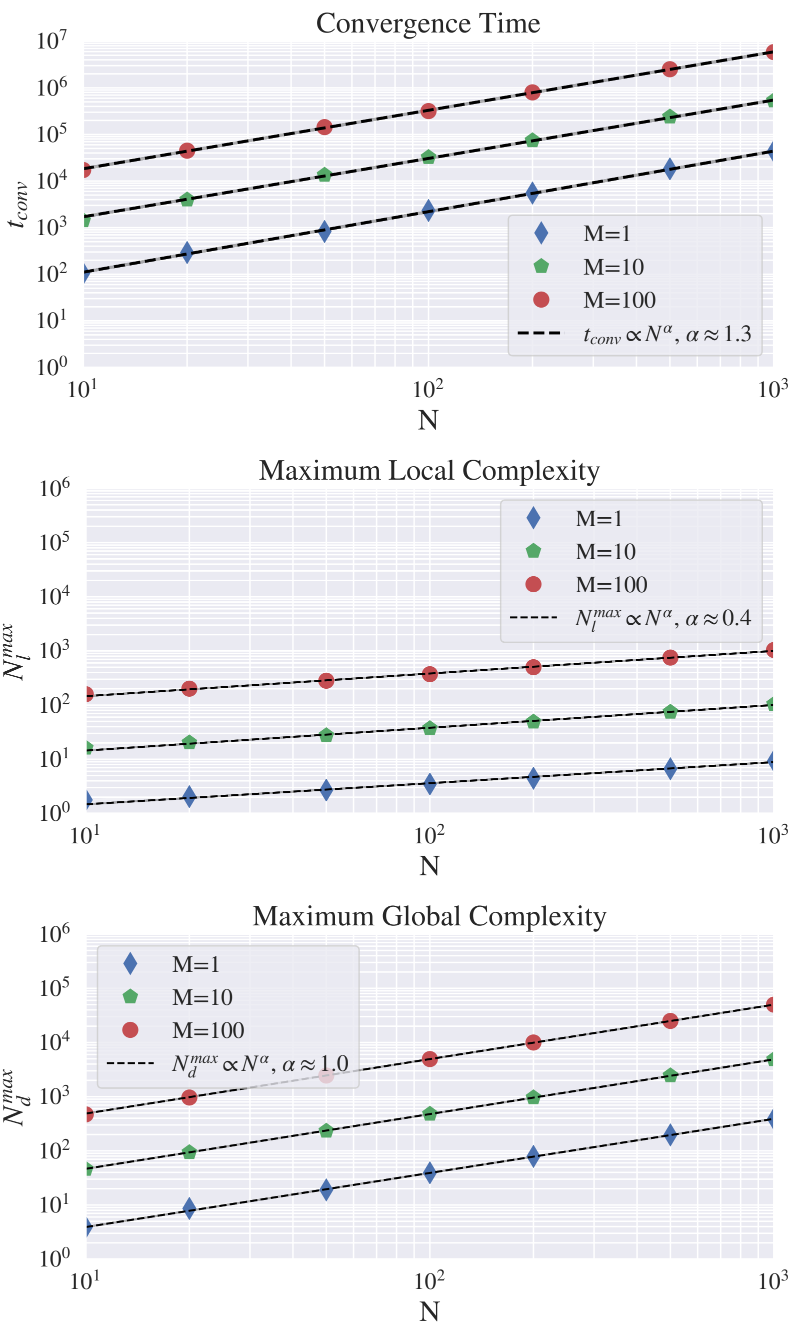


Figure 2.7: Scaling relations in the Naming Game, with respect to population size N. Convergence time t_{conv} and maximum local complexity N_l^{max} are proportional to $N^{1.5}$, and maximum global complexity N_d^{max} to N. With the standard Naming Game, all are directly proportional to M, which can be seen on the figures: all 3 curves are parallel and the shift between $M = 1$ and $M = 10$ is the same as from $M = 10$ to $M = 100$, and corresponds to a factor 10. It is this behavior that lead to study $M = 1$ only in previous studies. Each data point is an average over 8 simulations, and $W = \infty$. For further analytical details on the scaling laws, see chapter 5.

2.3 *Agent behavior*

An important part of any multi-agent system is the individual, or *microscopic*[13] behavior of an agent. In the Naming Game, individual agents can intervene at different points:

Topic choice: How they pick the topic m_s.

[13] By opposition to macroscopic, which is at population scale.

Word choice: How they pick a word w for this topic m_s using their lexicon.

Meaning guessing: How they guess a meaning m_h, provided a word w, again using the lexicon.

Success evaluation: How they evaluate the success or failure of the communication.

Vocabulary update: How they update their lexicons after the interaction, and under which conditions.

In practice, various *vocabulary update* policies have been studied and described (Wellens, 2012), while *word choice* and *meaning guessing* policies are usually depending on having scores in the vocabulary (i.e. values between 0 and 1, not necessarily binary.). A particular case of *word choice* policies have however been studied alongside the minimal NG (Baronchelli, 2006). *Success evaluation* is almost always the same: if the guessed meaning and the topic match: $m_S = m_H$ and the topic was already present in the hearer's lexicon. As for *topic choice* policies, they were first introduced in Oudeyer and Delaunay, 2008, and are the core feature that will be studied in this thesis.

2.3.1 Vocabulary Update Policy

At the end of each interaction, each agent takes into account the result of the interaction by modifying its lexicon. They do it on the basis of three elements: the topic m_S, the word w, and the failure or success of communication, a boolean $b_{success}$. There exists various policies that have been described and classified in Wellens, 2012. We will here review three of them: *imitation*, *minimal NG*, and *basic lateral inhibition*. In this thesis, only the first two will be used.

The update mechanisms are usually described for $M = 1$, and only synonymy appears as a potential source of conflict in lexicons. In our case, we will consider homonymy as well. Our choice was to deal with homonyms in exactly the same way as with synonyms. Computationally speaking, this can be considered an arbitrary choice, but experimental data suggests that human do regularize homonyms and synonyms in a symmetrical manner (Ferdinand and Spike, 2016).

Algorithm 1: **Imitation vocabulary update policy**, agents adopt the used conventions while removing potential synonyms and homonyms.

Require: Lexicon $\mathcal{A}$
1: **procedure** ImitationUpdate(m_S, w, $b_{success}$)
2: $\mathcal{A}_{m_S,w} \leftarrow 1$ ▷ Add used convention
3: $\mathcal{A}_{i,w} \leftarrow 0$; $\forall i \neq m_S$ ▷ Remove homonyms
4: $\mathcal{A}_{m_S,j} \leftarrow 0$; $\forall j \neq w$ ▷ Remove synonyms

Imitation vocabulary update policy is the simplest one: Agents adopt the word-meaning association that was used during the interaction; while removing any conflicting synonym or homonym. The speaker will only modify its vocabulary if the convention has just been invented. See algorithm 1. This strategy limits efficiently the memory that is used by each agent, but as it quickly erases information from past interactions (synonyms and homonyms), it also slows down the convergence process, which is the reason why it is usually not considered as a suitable update policy. See figure 2.9 for a comparison with other update policies.

Basic Lateral Inhibition strategy, or BLIS[14], is a strategy that is widely used, and relies on the lexicon using scores for each word-meaning association (with values between 0 and 1), that are reinforced or inhibited, depending on the outcome of the interaction (success or failure of communication). If the communication was successful, the score of the used association (m_S, w) is reinforced by a value δ_{inc} and potential conflicting synonyms and homonyms are inhibited by a value δ_{inh}. If the communication fails, the score of (m_S, w) is decreased by δ_{dec}. At creation of a convention, the score is set to s_{init}. Values do not leave the interval $[0;1]$.See algorithm 2. This policy is widely used as it allows faster convergence and is implicitly closer to what humans might be doing, as we do consider homonyms and synonyms. However, its dependence on four new parameters increases the complexity of studying the dynamics of the NG under various modifications.

[14] A table of symbols, acronyms and abbreviations can be found at the last page of this manuscript.

Require: Lexicon $\mathcal{A}$; parameters s_{init}, δ_{inh}, δ_{dec} and δ_{inh}

procedure BLIS(m_S, w, $b_{success}$)
  if $\mathcal{A}_{m_S,w} = 0$ **then** ▷ Add a new convention
    $\mathcal{A}_{m_S,w} \leftarrow s_{init}$
  else
    if $b_{success}$ **then** ▷ Successful communication
      $\mathcal{A}_{m_S,w} \leftarrow min(\mathcal{A}_{m_S,w} + \delta_{inc}, 1)$
      $\mathcal{A}_{i,w} \leftarrow max(\mathcal{A}_{i,w} - \delta_{inh}, 0)$; $\forall i \neq m_S$
      $\mathcal{A}_{m_S,j} \leftarrow max(\mathcal{A}_{m_S,j} - \delta_{inh}, 0)$; $\forall j \neq w$
    else ▷ Failed communication
      $\mathcal{A}_{m_S,w} \leftarrow max(\mathcal{A}_{m_S,j} - \delta_{dec}, 0)$

Algorithm 2: **Basic lateral inhibition vocabulary update**, agents reinforce or inhibit associations by small values. In particular, when communication is successful homonyms and synonyms are inhibited, but not deleted.

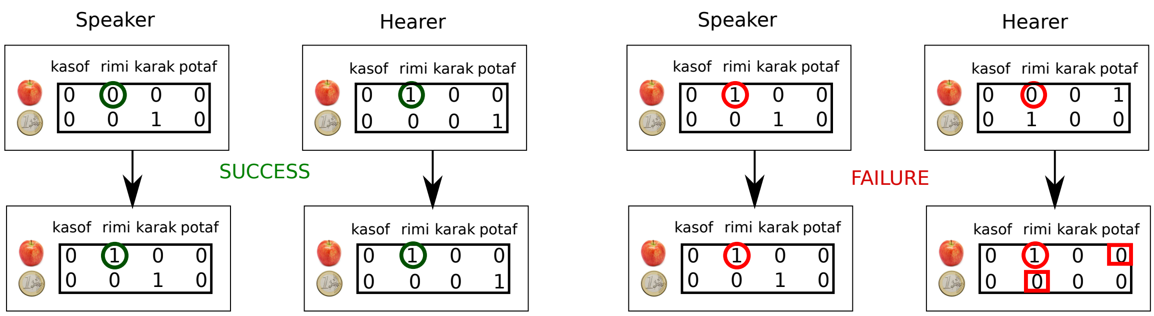


Figure 2.8: Illustration of the Minimal NG vocabulary update, in two cases: one successful communication, the other being a failure. In both cases, the association used by the speaker is added to the hearer's vocabulary. In addition to that, when the communication is successful, they both remove any competing homonymy or synonymy.

The minimal Naming Game update policy was introduced in Baronchelli, Felici, et al., 2006 as simpler than BLIS yet yielding similar dynamics. It is in fact a variation of BLIS with all parameters set to 1, i.e.: $s_{init} = \delta_{inc} = \delta_{inh} = \delta_{dec} = 1$. Its simplicity allowed an analytical approach to the Naming Game, which gave the scaling relations we saw in the previous paragraph, among other results. We can see in figure 2.9 that minimal NG and BLIS have similar dynamics. In the rest of this thesis, minimal NG will be used, unless in chapter 3.

Algorithm 3: **Minimal Naming Game update policy**, agents always add new conventions, but prune synonyms and homonyms when the communication is successful.

Require: Lexicon $\mathcal{A}$; parameters s_{init},δ_{inh},δ_{dec} and δ_{inh}

procedure MinimalNG(m_S, w, $b_{success}$)
    $\mathcal{A}_{m_S,w} \leftarrow 1$ ▷ Add convention
    if $b_{success}$ **then** ▷ Successful communication
        $\mathcal{A}_{i,w} \leftarrow 0$; $\forall i \neq m_S$
        $\mathcal{A}_{m_S,j} \leftarrow 0$; $\forall j \neq w$

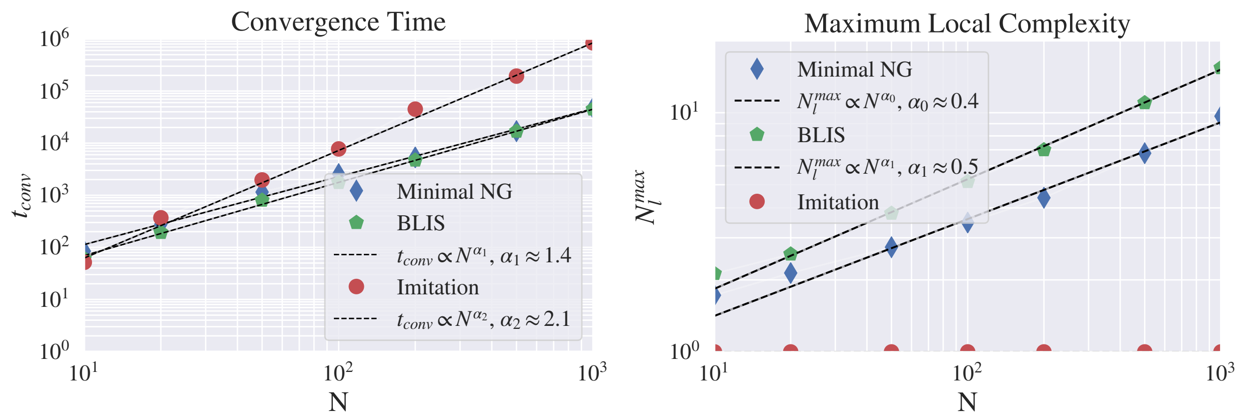


Figure 2.9: Comparison of different vocabulary update policies: imitation, minimal NG and BLIS. They all follow powerlaws, however BLIS and minimal NG have similar dynamics and have an exponent smaller than imitation. $M = 1$, $W = \infty$, averaged over 8 trials for each data point.

2.3.2 *Acceptance Policy*

A possible modification to the update policies presented in the previous paragraph is whether to accept or not the changes. The hearer might not trust the speaker, or the update would alter significantly the integrity of its lexicon for example. This mechanism, that we will call *acceptance policy* has been introduced in Baronchelli, 2006. So far, only stochastic choice was considered (depending on a probability β_{AP}, hearer will accept or not to perform the lexicon update). Such a mechanism slows down the convergence process, or even prevents it if β_{AP} is low enough. This two-fold behavior opens the path to modeling situations where convergence could happen or not depending on an external parameter[15], and was in this sense used in other models based on the Naming Game focusing on the emergence of contact languages (Pucci, Gravino, and V. D. P. Servedio, 2014; Tria, V. D. P. Servedio, et al., 2015).

[15] i.e. there will remain clusters of agents sharing different conventions.

2.3.3 *Topic Choice*

In the base model, the topic of each interaction is chosen randomly, with a uniform probability over the meaning space. But it is possible to change this to a non-uniform choice, made by the interacting

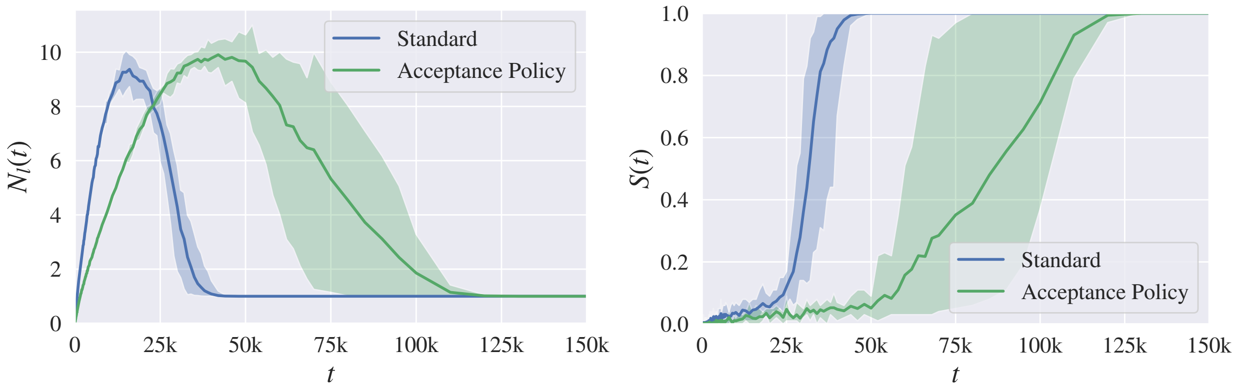


Figure 2.10: Typical dynamics when using probabilistic Acceptance Policy. The convergence process is slowed down, and the memory burst stays the same. $M = 1, N = 1000$, $W = \infty$ and $\beta_{AP} = 0.4$, averaged over 8 trials.

agents. Active topic choice has been introduced in Oudeyer and Delaunay, 2008, and similar mechanisms in another type of Language Game can be found in Cornudella and Poibeau, 2015. This very part of the Naming Game constitutes the core element that will be modified and studied along this thesis.

Usually the speaker mentally picks the topic, but there is an alternative possibility: the hearer can first choose the topic, and indicate it to the speaker, who will then consult its lexicon to find a corresponding word. This alternative scenario will be studied in chapter 3.

2.3.4 *Word Choice*

When the topic has been chosen (either randomly or via a certain policy), there can coexist several corresponding synonyms in the lexicon of the speaker, i.e. several words associated with an equivalent score to the meaning chosen as topic m_S. Which one should they choose as the word w that will be used in the conversation? In Baronchelli, 2006, three simple policies have been proposed:

Play last: Use the last word that was encountered as associated to m_S.

Play first: Use the oldest word still in the lexicon that was encountered as associated to m_S.

Play smart: A combination of both: *Play last* until having a successful interaction, than use the last word that triggered success[16].

[16] with a minimal NG update policy, it is equivalent to switching to *play first*.

Play last uses significantly less memory (powerlaw $N_l^{max} \propto N^{\alpha}$ of exponent $\alpha \approx 0.3$ instead of 0.5) but converges slower (to time scales that are not comparable with the standard NG). *Play first* converges faster (exponent $\approx$ 1.3 instead of 1.5 for t_{conv}) but memory usage stays high. *Play smart* combines advantages of both: less memory and faster convergence. This is the type of behavior that we are aiming at in this thesis; but we will see that this can be still ameliorated. Also, this approach cannot prevent a minimum burst of memory usage: a certain degree of synonymy is needed in the first place, to be able to choose between different words.

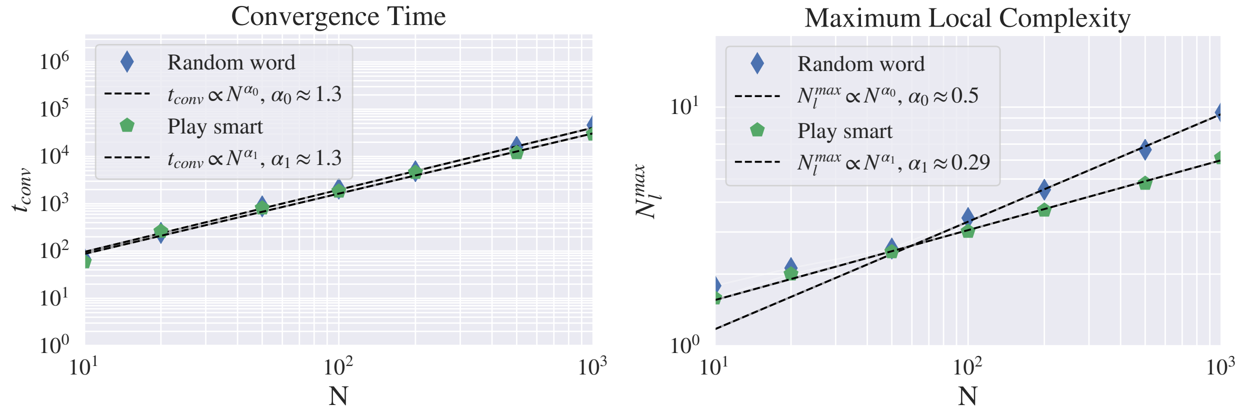


Figure 2.11: Comparison of two word choice policies: random choice and *play smart*. The latter has a lower memory peak, and converges a little faster (this behavior is amplified for $N > 10^3$, data not shown). $M = 1$, $W = \infty$, averaged over 8 trials for each data point.

2.4 *Population-level features*

Other variations of the Naming Game involve modifications not at the agent level, but in the structure of the population.

2.4.1 *Social network*

	Standard	Lattices ($d \leq 4$)	Networks
Max. memory N_l^{max}	$N^{1.5}$	N	N
Convergence time t_{conv}	$N^{1.5}$	$N^{1+\frac{2}{d}}$	$N^{1.4 \pm 0.1}$

Table 2.1: Scaling laws for population structured in social networks, depending on the population size N. Taken from Baronchelli, 2016.

Another feature that has been introduced in Baronchelli, 2006 is the notion of *social network*. In the real world, we do not interact with everybody else, but instead regularly interact with the same people, who themselves interact with their own network. It is our connections through other people which help spread information and new conventions. There are many possible structures for a social network: agents are elements of a line, or a 2D-grid, or are part of other types of networks with specific properties. The table below, adapted from Baronchelli, 2016, summarizes the contributions of such structures. Random networks[17]share the small-world property (each node is just a few hops away from any other) with the fully connected graph, and the low connectivity property (few neighbors per agent on average) with the low dimensional lattices. This ensures a trade-off between convergence time and memory.

[17] Different types of random graphs exist, having several properties. See Barabási and Albert, 1999 and Erdos and Rényi, 1960.

2.4.2 *Population turnover*

Real populations of language speakers are not static: new individuals are introduced regularly, either by birth or by simple arrival in a new country. Symmetrically, some might disappear from the community. The impact of adding and removing agents from the population of agents in the Naming Game has been studied in Steels and Kaplan, 1998 and Vogt and Coumans, 2003. Typically, every $T_r = t_r \cdot N$ interactions, one agent is removed, and a new one is introduced, having a completely empty lexicon. The main result is that over a certain speed of turnover ($s_r = \frac{1}{T_r}$), the lexicon becomes unstable and the proportion of successful communications drops. See figure 2.13.

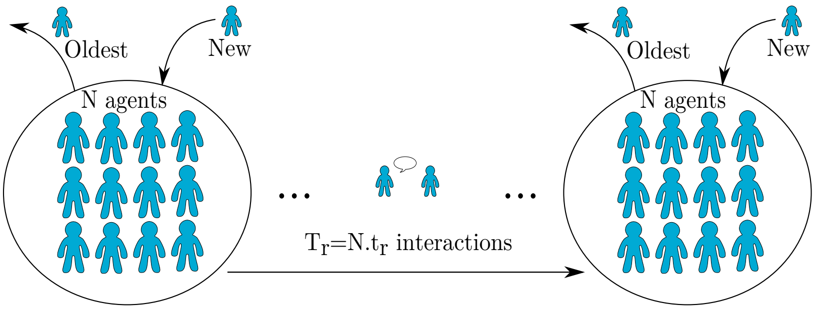


Figure 2.12: Illustration of the population turnover mechanism

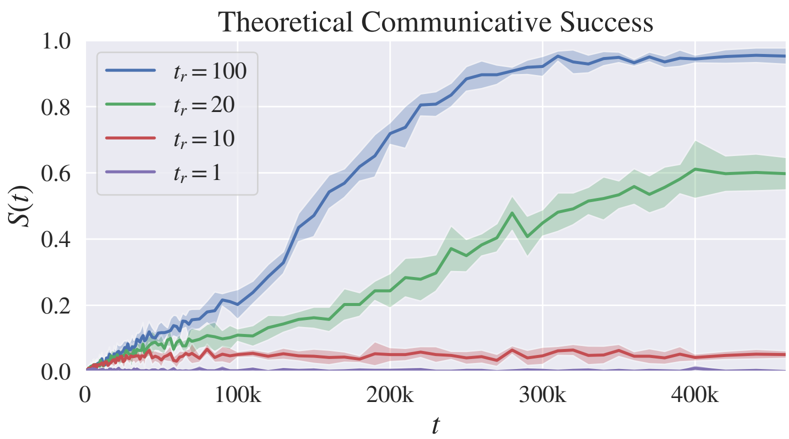


Figure 2.13: Impact of population turnover on global dynamics, depending on the replacement rate. The slower the speed of turnover (higher values of t_r), the higher is the final stable agreement level reached by the population. $M = N = 100$, $W = \infty$, averaged over 8 trials.

2.5 *Discussion*

2.5.1 *Classification and standard model definition*

As a conclusion to this chapter, we propose a classification of the Naming Game models, represented in figure 2.14. A slightly more detailed version, used for the implementation of all simulations presented in this thesis, can be found in appendix A. The description of the standard Naming Game in the classification is the following:

Agent picking/social network: Full-connected graph; speaker and hearer are picked from the population, with a uniform probability over all possible couples.

Population evolution/agent replacement: Nothing happens between interactions.

Interaction scenario: The speaker refers to the topic using an associated word from its vocabulary or by inventing a new association, the hearer interpretes the uttered word as a meaning using its own vocabulary. See figure 2.3.

Topic choice strategy: Random topic choice (uniform probability over the set of meanings).

Vocabulary update policy: Minimal NG.

Starting condition: Empty lexicon (all-zero matrices)

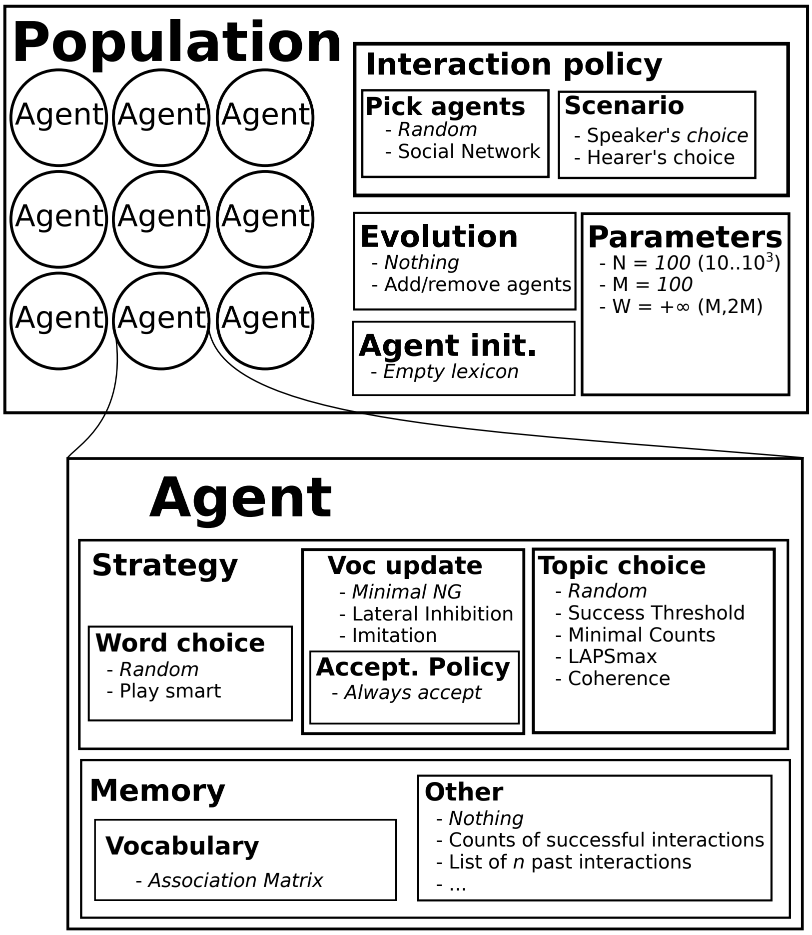


Figure 2.14: Modular representation of the NG models used in this thesis, with different possibilities for each module. The first possibility for each module — also in italic — is the one used in the standard model.

2.5.2 *Issues of the Naming Game*

While going through this taxonomy of variants of the Naming Game, we have identified a few key elements of the model that one can seek to improve:

Memory peak: Individual agents need more memory than the final lexicon, and it increases with population size. Memory usage could be decreased.

Convergence: Value and dependence to N (powerlaw) of time to full convergence could be decreased.

Unstability with replacement: Over a certain rate, population turnover makes the system unstable. The critical rate r_c at which it starts to happen could be increased.

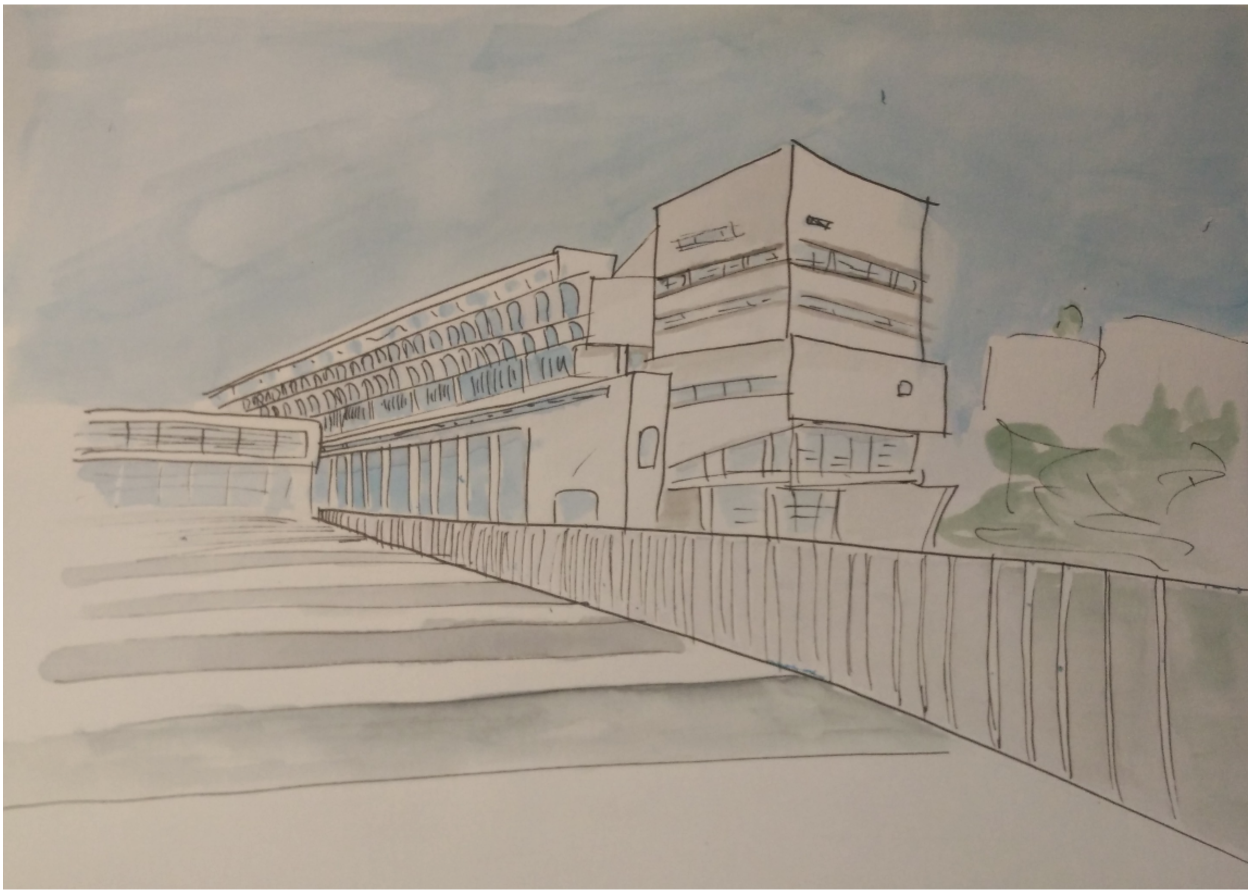
Diffusion on a graph: Certain topologies of social networks are associated with slow diffusion of conventions.

We have seen that the typical dynamics of the Naming Game involve a transition from a first phase of rapid complexity growth, to a slower second phase of pruning most of the conflicting conventions. Some of the proposed variants of the Naming Game do reduce the complexity peak, but still go through a phase of growth, which is the main reason to the previously mentionned issues. Is it possible to mix both phases to limit the complexity growth and/or converge faster? We will see that active policies, especially topic choice, are an adapted tool to address this problem.

Como Summer School, Italy. September 2016

Part II

Active Topic Choice



ICDL-Epirob Conference, Cergy. October 2016

3
Active choice under constrained memory

Contents

3.1 *Introduction*

In the standard Naming Game, the interaction scenario (see section 2.1.4) starts with the speaker picking one of the available meanings as a topic for the conversation that is about to take place. However, this was done so far using a uniform random pick over the set of meanings $\mathfrak{M}$. This assumption was even the root justification for dropping additional meanings and narrowing down $\mathfrak{M}$ to a singleton in previous analytical work (Baronchelli, 2006). It is a strong assumption, as humans typically do not choose their topic of conversation or their words in this way, but show biased patterns in their choices (Kachergis, Yu, and Shiffrin, 2013; Zipf, 1949).

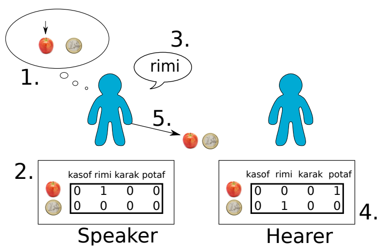


Figure 3.1: Reminder of the standard interaction scenario: The first step is an internal choice of a topic by the speaker. For the detailed description see paragraph 2.1.4.

By actively choosing the topic of their conversation, people can select which conventions they want to disambiguate, spread among their peers, or learn about. This behavior can be introduced in the model with a simple change: during the first step of the interaction scenario, the meaning chosen as topic is not picked randomly, but according to a given policy. This policy can only rely on information locally available to the agent making the choice, and not global information at population level. To stay cognitively plausible, the computational cost of policies should remain low.

In this chapter, we will study the effect of Active Topic Choice on the dynamics of the Naming Game under memory constraints. In such a scenario, we can artificially force the complexity burst not to happen, but can we still get a fast convergence towards a stable lexicon? We will first focus on really strong constraints: only the current state of the lexicon is known, and both homonymy and synonymy are forbidden. Secondly, we will extend memory to a count of the successes of past interactions. Various strategies will be presented for each case. All rely on an exploration vs. exploitation paradigm.

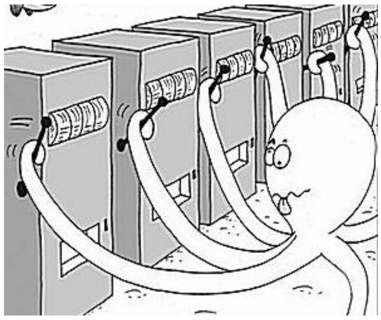


Figure 3.2: Illustration of a Multi-Armed Bandit. Slot machines are also called *One Armed Bandit*, what if you can choose different arms which give different rewards? You have to balance between using the best arm so far, and exploring/trying new ones. Image source: microsoft.research.com

3.1.1 Exploration vs. Exploitation

Exploration vs. exploitation is a paradigm commonly used in machine learning. When faced with a choice between different options (which restaurant should I pick tonight?), one can either choose the best option so far (I always go to the italian place, their pizzas are good!), or gather more information by trying a new option (Maybe I could go to this new restaurant, or to that other one where I tasted only one meal?). Gathering information is essential, as there may be better options among the unexplored ones. Another common example is a set of slot machines with different reward distributions, unknown at the beginning: The more you try each of them, the better you approximate those distributions and the better you can choose between them to get the best reward over time. This analogy has been formalized mathematically, and is called the Multi-Armed Bandit problem. It has been studied extensively and many algorithms have been designed to solve it (Bubeck, Cesa-Bianchi, et al., 2012). For the moment we will focus on the description of our problem within the explore/exploit paradigm, but we will come back to Multi-Armed Bandit algorithms (or MAB[1]) in chapter 4.

Where in our problem do we find exploration? When an agent decides to pick a new topic, a meaning that he never spoke or heard of before. In other words, when an agent invents a new convention, a new meaning-word association. This leads necessarily[2] to a failure in communication, as the new convention is just being taught to the hearer; but it is a step towards completion of the vocabulary. The alternative, or exploitation, is to choose a known meaning: one can expect to get successful communications because the chosen convention is already present in the rest of the population, while at the same time reinforcing the convention and spreading it further into the population. Also, when several words are associated to the

[1] A table of symbols, acronyms and abbreviations can be found at the last page of this manuscript.

[2] Excepted when the number of possible words is finite and the same convention has already been invented by another agent and taught to the hearer, but this event stays unlikely in most cases.

chosen meaning, exploitation is a way to settle on only one of the synonyms and discard the others, depending on the outcome of the interaction. Homonymy, when it exists, can also be disambiguated in this manner.

The reward is in our case unclear[3], but we can already assess the necessity of balancing the two behaviors: if the interacting agents never explore, there will be no complexity bursts[4], but if some of the meanings have not been used in an interaction, they will never be. Exploiting too much therefore limits memory usage, but leads to a dead-end where the lexicon is never built in its entirety. On the other hand, only exploring introduces by definition numerous new conventions, at least more than with random topic choice, hence a higher complexity burst, and a slower convergence process as there are many more conflictual conventions[5]. The main contributions in this thesis deal with finding an adapted algorithm to effectively balance the two behaviors.

[3] We will propose a reward function in chapter 4

[4] There can be one if $N \gg M$, but it will be anyway smaller than with random topic choice. See chapter 5.

[5] i.e. a higher degree of homonymy and synonymy

3.1.2 *Memory constraints*

The flow of information observed by an agent consists of the topic of the interaction, the word used to refer to the topic, and whether the communication was successful or not. Using this information, each agent shapes incrementally its own lexicon. But it is also possible to store the information of the past interactions in an another way, preferably not all the information but a small processed representation of it[6], and use it in for their topic choice policy.

[6] Reminder: we are trying to decrease memory usage, if we allow to store a lot of information to keep the lexicon small, that will still be memory usage.

The minimum memory needed to build a stable lexicon, is obviously the size of a completed lexicon, M[7]. There is a way to constrain the system not to go over this value: forbid synonymy and homonymy. There would be at maximum one word per meaning, which effectively constrains the local complexity $LC(V) \leq M$.

[7] The memory measure that is used is the number of associations of a lexicon, also called local complexity, described in section 2.2.3

The imitation vocabulary update, described in section 2.3.1, is ensuring exactly what we are searching for: potential homonyms and synonyms are immediately dropped at first exposition with conflictual information. In all simulations presented in this chapter, we will use this vocabulary update policy, apart from the necessary comparisons to the original model. There is still convergence with imitation and random topic choice, but the dynamics are extremely slower, which is the price paid by the limits imposed to memory. The first class of policies that we will present in this chapter do not use any more information than the lexicon itself. The memory usage is therefore always smaller than M.

Information gathered during interactions could however be really useful for a topic choice policy. A natural and low-memory information is the number of successful and failed interactions for each meaning. It scales directly with M [8], and therefore with the minimal memory M. We will use this information in the second part of this chapter.

[8] Maximum value: M (lexicon) $+2M$ (successes and failures)

3.1.3 *The two levels of an ATC policy*

Choosing a topic is picking a meaning among the set of available meanings $\mathfrak{M}$. However, This set can be divided into two subsets: The set of known meanings $\mathfrak{M}_k$ which have at least one word associated to them in the lexicon, and the set of unknown meanings $\mathfrak{M}_u$ which have no words referring to them yet.

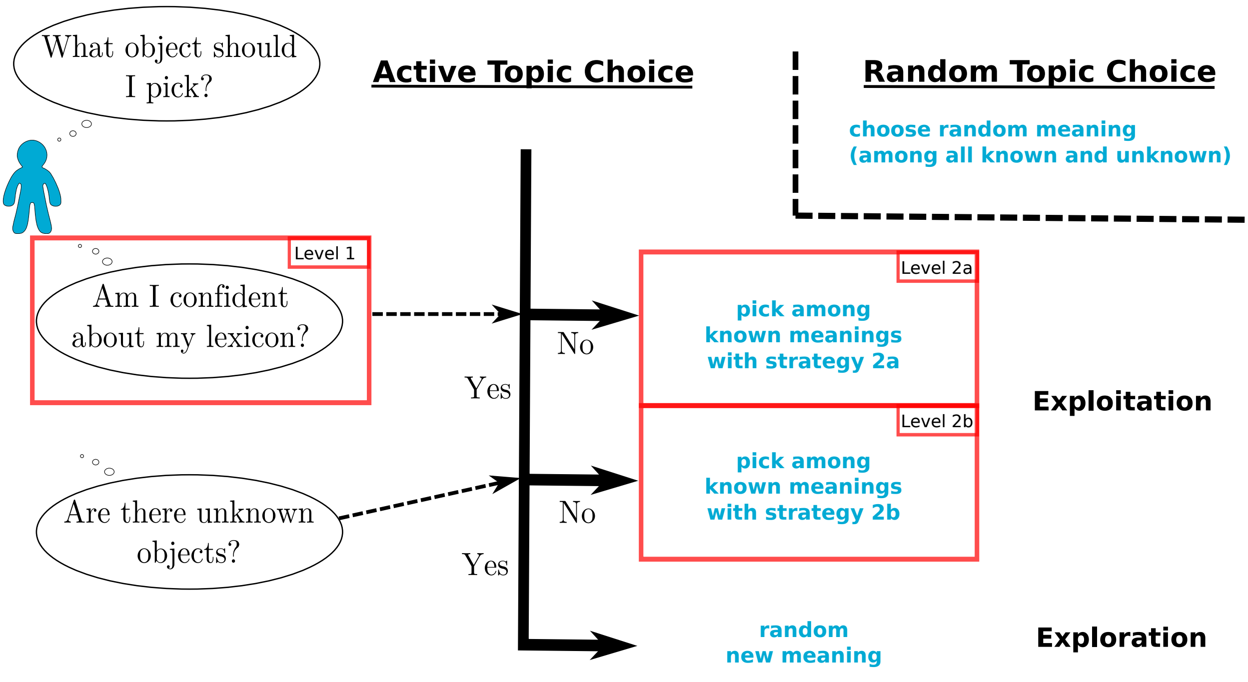


Figure 3.3: Illustration of the two levels of ATC policies. First level: Exploration vs. Exploitation. Second level: Choosing among known meanings, when there are still meanings left to explore (2a) or when there aren't (2b).

As we have seen in paragraph 3.1.1, the first and main level of an ATC policy lies in choosing whether to explore or exploit, i.e. picking a topic from respectively $\mathfrak{M}_u$ or $\mathfrak{M}_k$. All unexplored meanings are by definition equivalent, and when exploring, one cannot further refine the decision and can only pick one randomly. However, when exploiting, different meanings may have different histories: One can be known for a long time and having yielded many successes in the past, when others can be brand new and with little associated information. The second level of an ATC policy consists in choosing which known meaning to pick when exploiting. In the first part of this chapter however, because of the hard memory limitations one cannot have any information differentiating the known meanings. The second level will only be explicitely used in the second part of the chapter. It can be noted that the second level can itself be split into two parts: during the learning phase where there are meanings left to explore, and during a pure negotiation phase where all meanings have already been explored. When the strategy will be different in those two situations, they will denoted as respectively level 2a and level 2b.

Both levels rely on a notion of confidence about known meanings: did their associated conventions spread enough in the population, or do I still need to reinforce them? Defining each policy will be equivalent to defining the confidence measure that will be used, along with a potential threshold of when to shift behavior.

3.2 *Hard constraint on memory*

The first class of policies uses only the lexicon itself as information. Because we are using the imitation vocabulary update[9], and because all meanings are supposed equivalent, we can only intervene on the first level of the topic choice policy (exploration vs. exploitation).

[9] See previous paragraphs for a justification of this choice, and section 2.3.1 for details.

3.2.1 *Decision Vector policies*

Algorithm 4: **Decision Vector Topic Choice**, function of the total meaning space $\mathfrak{M}$ and the subset of known meanings $\mathfrak{M}_k$, given a vector of size $M+1$. Every strategy with this level of memory constraints can be reduced to such a vector.

Require: Vector $(D_i)_{i \in [\![0\ ;\ M]\!]}$ **Notation:** $\mu = |\mathfrak{M}_k|$

procedure DecisionVectorTC($\mathfrak{M}, \mathfrak{M}_k$)
  with probability D_μ:
    $m \leftarrow$ sample uniformly from $\mathfrak{M} \setminus \mathfrak{M}_k$ ▷ Explore
    m is added to $\mathfrak{M}_k$
  with probability $1 - D_\mu$:
    $m \leftarrow$ sample uniformly from $\mathfrak{M}_k$ ▷ Exploit
  return m

The only useful information available is the size of the lexicon, with values in $[\![0\ ;\ M]\!]$. As stated at the beginning of this section, with this level of memory constraints the choice is reduced to two options: exploring or exploiting. Because of this, in each possible state the decision policy can be reduced to the probability of exploring. In other words, the behavior can be represented by a vector of dimension $M+1$, where each coordinate (from 0 to M) would be the probability to explore, if found as a speaker with the corresponding number of known meanings. The first coordinate is necessarily 1 and the last necessarily 0, because respectively exploitation and exploration are not possible in these cases. See algorithm 4. All algorithms in this section will be given their representative decision vector, as their signature.

An important remark is that 0 as a value in the vector should be avoided, because it could lead to blocked states. In fact, if all agents have a stable and shared vocabulary of this size, they will never explore, and their lexicon will never grow and get completed. See figure 3.4 for an illustration.

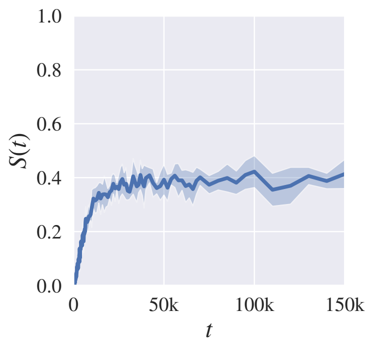


Figure 3.4: Illustration of a blocked state. Agents do not reach full agreement because they stopped exploring: they only exploit. They reach an agreement on the explored subset of meanings, but will never even have a word for the other ones. $M = N = W = 100$, averaged over 8 trials. Oscillations come from the way of computing $S(t)$: via Monte Carlo sampling.

3.2.2 *Exploration bias*

A first strategy is to consider *exploration* as an equivalent option to any other known meaning, as an *alternative meaning*. The exploration probability decreases over time, inversely proportional to the size of the set of known meanings $\mathfrak{M}_k$. See algorithm 5. This process can be related to another class of stochastic processes, explained in the following paragraph.

Algorithm 5: **Exploration-biased Topic Choice**, function of the total meaning space $\mathfrak{M}$ and the subset of known meanings $\mathfrak{M}_k$. When possible, exploring is inversely proportional to the number of known meanings $|\mathfrak{M}_k|$. This is similar to the *Chinese restaurant process*, with a modification on the weights. See main text for a more detailed explanation.

procedure ExploBiasedTC($\mathfrak{M}, \mathfrak{M}_k$)
  if $|\mathfrak{M}_k| < |\mathfrak{M}|$ **then** ▷ Are there unexplored meanings?
    with probability $\frac{1}{|\mathfrak{M}_k|+1}$:
      $m \leftarrow$ sample uniformly from $\mathfrak{M} \setminus \mathfrak{M}_k$ ▷ Explore
      m is added to $\mathfrak{M}_k$
    with probability $\frac{|\mathfrak{M}_k|}{|\mathfrak{M}_k|+1}$:
      $m \leftarrow$ sample uniformly from $\mathfrak{M}_k$ ▷ Exploit
  else
    $m \leftarrow$ sample uniformly from $\mathfrak{M}$
  return m

Corresponding Decision Vector:
$D_i = \frac{1}{i+1}$; $i \in [\![1\ ;\ M-1]\!]$.

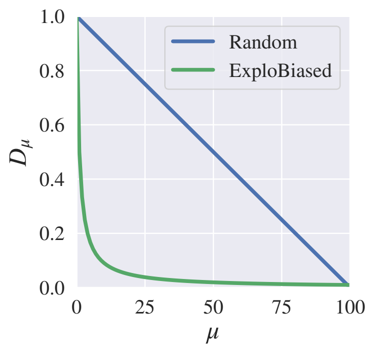


Figure 3.5: Decision Vector for the Exploration-biased strategy, $M = 100$.

The Chinese restaurant process (Aldous, 1985), is a metaphor for a decision process between a growing number of options. Customers are coming one by one to a restaurant having an infinite number of round tables. Each one of them has the choice to sit at one of the already occupied table, or at an empty table. Probabilities of these choices are weighted with the number of occupants at the table, or with a parameter α for the choice of an empty table. After n customers entered the restaurant, the probability of choosing an empty table is therefore $\frac{\alpha}{\alpha+n}$.

We can see the Exploration Biased strategy as a Chinese restaurant process. In our case, meanings $\mathfrak{M}$ are tables, and occupied tables would be known meanings $\mathfrak{M}_k$. As we cannot count the *sitted customers* for each meaning, we keep it equal to 1 for each table. We also consider $\alpha = 1$.

3.2.3 *Information Gain maximization*

The process of convergence is towards one of the many possible completed lexicons. At the beginning, all $M! \cdot (W-M)!$[10] completed lexicons are equiprobable as outcome of the negotiation process. But in intermediary states, particularly close to convergence, many of the possible lexicons become improbable or even can be discarded. From one state to the other, because the probability distribution over the possible outcomes changes, there is a gain of information.

[10] If W is pseudo-infinite, one can consider $W = M \cdot N$, which is an upper bound to the maximum number of inventions.

At an individual level, not being in a locally converged state (i.e. a completed vocabulary), means that the agent *doesn't know yet* what the conventions would be in the completed state, among all the remaining possibilities. In other words, it lacks a certain quantity of information to build a complete vocabulary. What is this missing information, in bits, needed by an agent to fill its association matrix? If we have M available meanings and W available words ($W \geq M$), the number of possible configurations for a completed vocabulary is:

$$\Omega = \frac{W!}{(W-M)!} \tag{3.1}$$

Therefore, the information needed, in bits, to define such a vocabulary among all possible ones is:

$$I(M, W) = \log_2 \Omega = \sum_{k=0}^{M-1} \log_2(W-k) \tag{3.2}$$

Now, if we consider an intermediate state, in which an agent already has a vocabulary of μ associations[11], without synonyms or homonyms, we can also calculate the information missing in this case. It is quite simple, as the already known meanings and words do not matter anymore in the calculation: By removing them, we get back to the initial problem, i.e. evaluating the missing information of an empty vocabulary, now with $M - \mu$ available meanings and $W - \mu$ available words:

$$i(\mu, M, W) = I(M-\mu, W-\mu) = \sum_{k=\mu}^{M-1} \log_2(W-k) \tag{3.3}$$

However, maximizing this information on the individual vocabulary of the speaker is straight-forward: by always exploring, one can quickly reach a completed lexicon. But this does not guarantee that this lexicon is shared by the others or compatible with their lexicons. The hearer's lexicon evolution on the contrary is less well-known: whatever the state of the speaker's lexicon and the topic that was chosen, μ_H[12] can still increase, decrease or remain the same; the possible situations and associated outcomes are:

- $\mu_H \leftarrow \mu_H + 1$: The used convention was not present in the lexicon, and is not in conflict with any other convention in Hearer's vocabulary.
- $\mu_H \leftarrow \mu_H$: Hearer already had the convention, or while adding the convention another one was removed, synonym or homonym.
- $\mu_H \leftarrow \mu_H - 1$: While adding this convention to the lexicon, both a synonym and an homonym had to be removed.

It makes sense to try to influence this evolution by the choice of the topic: given a certain probability distribution over the possible lexicons of Hearer, the speaker can compute the probabilities of each outcome, and an expected lexicon change $\Delta\mu_H$ and associated information gain G_μ for the hearer. But what can be its guesses about the state of the lexicon of the hearer? What distribution can be considered over the space of possible lexicons? The most probable value for μ_H is μ_S, because both agents are supposed to have had the same

[11] i.e. $\mu = |\mathfrak{M}_k|$, this notation will be used often in the rest of this chapter, also extending it to hearer's lexicon size μ_H and speaker's μ_S

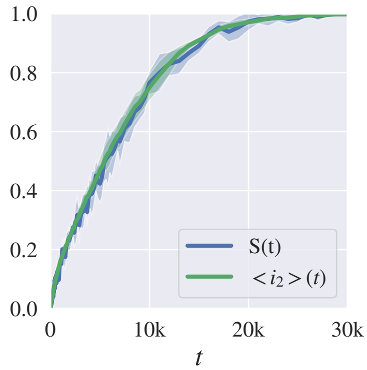


Figure 3.6: Comparison of TCS measure $S(t)$ and normalized shared information $< i_2(t) >$. $N = 10$, $M = 100$, $W = 200$, averaged over 8 trials.
Interesting remark: we can consider the information of the shared vocabulary between a couple of interacting agents, constituted of only the word-meaning association present in both lexicons. Averaging this value over all possible couples of agents can be a good approximation of the TCS, see figure above for empirical results.

[12] Size of Hearer's lexicon

number of past interactions. As for its composition, the safest assumption (i.e. a subestimation of the information) is to consider that both lexicons have been built independantly so far, and any shared association would be so by chance[13].

With the two hypothesis $\mu_H = \mu_S$ and independance of lexicons, we can derive the probabilities of the different OUTCOMES FOR THE OTHER AGENT (in terms of change of lexicon size) depending on the size of the lexicon μ and the two possible choices, exploring or exploiting. This can be done from the perspective of both agents, speaker and hearer. The probabilities are summed up in table 3.1. For the moment, only values for Speaker are relevant, but the values for Hearer will be used in paragraph 3.4.3, in an alternative scenario where Hearer can pick the topic.

[13] This is particularly true if $N \gg M$.

Speaker's perspective		
$\Delta\mu$	Info. Gain G_μ (bits)	Probability p_μ
$+1$	$\log_2(W-\mu)$	$\frac{(M-\mu)(W-\mu)}{MW}$
0	0	$\frac{\mu(1+M+W-2\mu)}{MW}$
-1	$-\log_2(W-\mu+1)$	$\frac{\mu^2-\mu}{MW}$

Hearer's perspective		
$\Delta\mu$	Info. Gain G_μ (bits)	Probability p_μ
$+1$ (explore)	$\log_2(W-\mu)$	$\frac{W-\mu}{W}$
0 (explore)	0	$\frac{\mu}{W}$
0 (exploit)	0	$\frac{W-\mu+1}{W}$
-1 (exploit)	$-\log_2(W-\mu+1)$	$\frac{\mu-1}{W}$

Table 3.1: Information outcomes with associated probabilities from both Speaker and Hearer's point of view, when having μ associations in their vocabulary. There are M meanings and W words. To determine the probabilities, each agent assumes the other agent's vocabulary to be a permutation of its own or in other words, that they share the same μ value. Information measure (defining the gain) is introduced in the main text.

THE ASSOCIATED GREEDY STRATEGY is to choose the option (explore or exploit) associated with the highest expected outcome. This would however lead to a decision vector comporting several times 0. As discussed earlier, this can lead to blocked states, as depicted in figure 3.4. An alternative is to use a *softmax* function[14]: see equation 3.4.

For the hearer however, the decision would be heavily biased towards exploration at the beginning, hence the chosen function does not reflect a maximization of the information gain, but a minimization of information loss and an avoidance of early exploration (see equation 3.5).

[14] Softmax functions are used to smooth decision processes by not selecting the highest rewarding option, but sampling from the different options using a probability distribution depending on the expected rewards. They usually rely on a parameter, here noted β, called temperature parameter, which comes from an analogy with physics.

Speaker:

$$P_{exp}(\mu,\beta) = \frac{e^{\frac{G(+1)\cdot p(+1)}{\beta}}}{e^{\frac{G(+1)\cdot p(+1)}{\beta}} + e^{\frac{G(-1)\cdot p(-1)}{\beta}}} \tag{3.4}$$

Hearer:

$$P_{exp}(\mu,\beta) = \frac{e^{-\frac{G(-1)\cdot p(-1)}{\beta}}}{e^{\frac{G(+1)\cdot p(+1)}{\beta}} + e^{-\frac{G(-1)\cdot p(-1)}{\beta}}} \tag{3.5}$$

If we look at the shape of the vector (see figure 3.7), it corresponds to *exploring until a certain lexicon size*. In other words, exploring enough per agent so that everybody would end up with a full lexicon, but not too much to introduce as few conflicts as possible. The optimal size for switching is the value of μ for which $G(+1) \cdot p(+1) = G(-1) \cdot p(-1)$. Solving the equation gives $\mu \approx \frac{MW}{M+W-1}$.

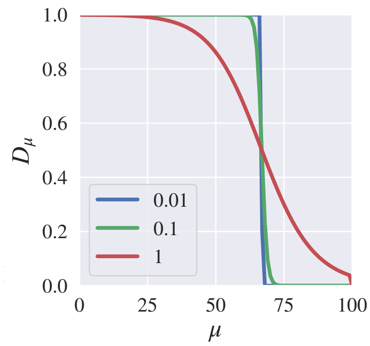


Figure 3.7: Decision Vector for the Information Gain maximization strategy, for several values of parameter β. $M = 100$, $W = 200$.

Information Gain maximization Decision Vector:

$D_i = P_{exp}(i, \beta)$; $\quad i \in [\![1\ ;\ M-1]\!]$.

3.2.4 *Chunks Decision Vector strategy*

Another strategy is to explore by *chunks* instead of considering the whole set of meanings. Everybody explores only once, and when the expected size of lexicon resulting of the spreading of these inventions is reached, explore again, until reaching the next threshold. The expected number of chunks $n_{ch}(M, N)$ and the respective remaining lexicon sizes M_i are studied in section 5.2.4.

Because there is a small probability that the system does not reach n_{ch}, but only $n_{ch} - 1$ for example, it is necessary to allow exploration at every step with a residual probability γ_{ch}. We will typically use $\gamma_{ch} = 0.001$, or $\gamma_{ch} = 0.01$for low values of N.

Algorithm 7: **Chunks Topic Choice**: Representation of the strategy as a decision vector. This strategy explores by *chunks* of meanings, whose size is guessed from N, M and W.

Chunks Decision Vector:

Require: $(M_j)_{0 \le j \le n_{ch}}$

$D_i = 1$ where $M - i \in (M_j)_{0 \le j \le n_{ch}}$; γ_{ch} elsewhere ;

$i \in [\![1\ ;\ M-1]\!]$.

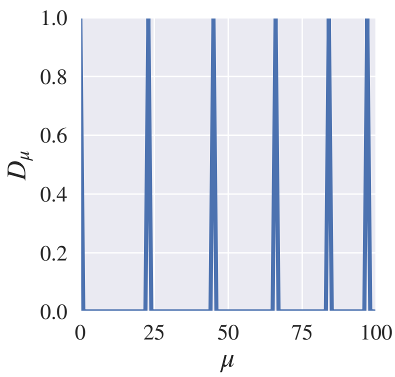


Figure 3.8: Decision Vector for the Chunks strategy, $N = 50$, $M = 100$, $W = 200$.

3.2.5 *Comparison to optimal vector*

For low values of M, we can use classic optimization algorithms to try to find an optimal shape of the decision vector, minimizing the convergence time. On figure 3.9, we can see the resulting shape compared to the Chunks decision vector. Further optimizations and fine-tuning with different algorithms may be necessary to ensure that this vector is indeed optimal, but at least we can see that this vector is closer to the Chunks strategy than to the Information Gain maximization strategy: it favors multiple waves of exploration instead of a single wave stopping at a certain lexicon size. An illustration of the better convergence properties of Chunks (compared to Info. Gain maximization and Exploration Biased) can be seen on figure 3.10. For later comparisons with other strategies, we will only keep the Chunks strategy, having the best performance so far.

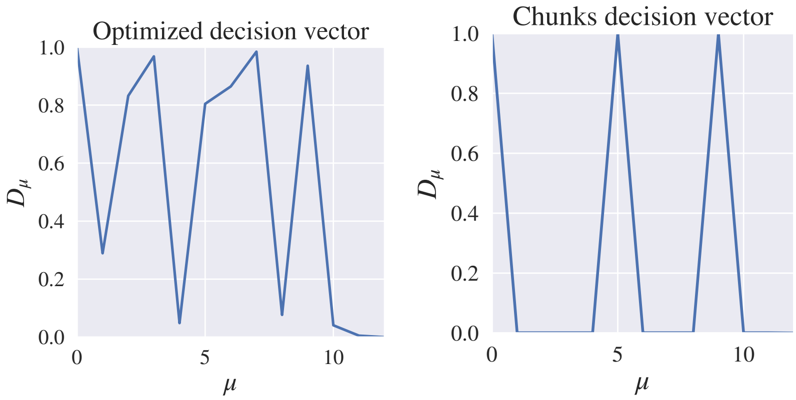


Figure 3.9: Optimized decision vector compared to chunks decision vector: $N = 10$, $M = 12$, $W = 24$. Optimization algorithm: CMA-ES, initialization parameters: vector: 0.5 everywhere; σ : 0.5.

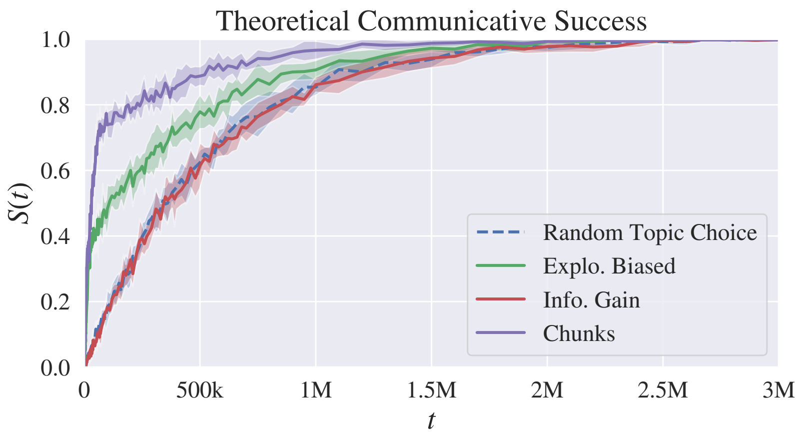


Figure 3.10: Comparison of the dynamics of Info. Gain, Explo. Biased and Chunks strategies. Info. Gain has similar dynamics has RTC. Chunks and Explo. Biased starts faster, but the agreement on the last meanings is slower and the final convergence time is just slightly smaller. $M = 100$, $W = \infty$, averaged over 8 trials.

3.3 *Counting successful interactions*

Considering past interactions, and in particular successes, one could probably build a more efficient strategy. This idea has been introduced in Oudeyer and Delaunay, 2008, where the measure of confidence triggering exploration is the ratio of successful events among past interactions. We will redefine a strategy based on this measure, called *Success Threshold*, and another one based on the absolute count of successes, *Minimal Counts*. Because each individual known meaning m carries a different past, it is possible to distinguish them, and therefore add a second level to the strategy, as defined in paragraph 3.1.3.

3.3.1 *Success Threshold*

The Success Threshold strategy (see algorithm 8) uses the ratio of past successful interactions: an agent should explore when its value gets over a threshold parameter $\alpha_{ST} \in [0 ; 1]$. The success ratio S_R is defined as the mean of the success ratio over known meanings:

$$S_R(t) = \langle S_R(m,t) \rangle_{m \in \mathfrak{M}_k} \tag{3.6}$$

If $S_R(t) \leq \alpha_{ST}$ (level 2.a in the description found in paragraph 3.1.3), we will choose the meaning having the smallest success ratio, hence presenting the most room for improvement.

If all meanings have been explored (level 2.b in the description found in paragraph 3.1.3), i.e. $\mathfrak{M}_k = \mathfrak{M}$, we will fall back on a random topic choice, this will be discussed at the end of the chapter.

Algorithm 8: **Success Threshold strategy**, exploring when the ratio of past successes per meaning goes over a value α_{ST}.

Require: Success ratios $S_R(t)$ and $S_R(m,t)$; parameter α_{ST}
procedure SuccessThresholdTopicChoice($S_R(m,t)$; $m \in \mathfrak{M}_k$)
  if $S_R(t) \leq \alpha_{ST}$ **then** ▷ Level 1
    if $|\mathfrak{M}_u| > 0$ **then**
      Pick random m from $\mathfrak{M}_u$ ▷ Explore
    else
      Pick random m from $\mathfrak{M}$ ▷ Level 2.b
  else
    Pick $m = \arg\min_{m \in \mathfrak{M}_k} (S_R(m,t))$ ▷ Level 2.a

To set the parameter α_{ST}, we can optimize the convergence time with fixed values of N and M. A minimum is found around $\alpha_{ST} = 0.8$, as can be seen on figure 3.11. On the same figure, we can see that both levels are needed to achieve significantly faster convergence than with Random Topic Choice, by comparing with convergence times associated with a modified strategy with only level 1. For further uses of this strategy, α_{ST} will be set to 0.8, which appears to be stable for different values of N ($N \in [10, 100, 1000]$, data not shown.).

3.3.2 *Minimal Counts*

The Minimal Counts strategy (see algorithm 9) is similar to Success Threshold, but uses an absolute count of successful interactions instead of relative. We will change another detail: we do not consider the mean value of success counts per meaning $< S_C(m,t) >$ but the minimum $\min(S_C(m,t))$. The threshold parameter will be noted n_{MC}. This allows to keep an important weight for new meanings with very few interactions, which was anyway the case with Success Threshold.

Again, to set the parameter n_{MC}, we can optimize the convergence time with fixed values of N and M. We will consider a normalized

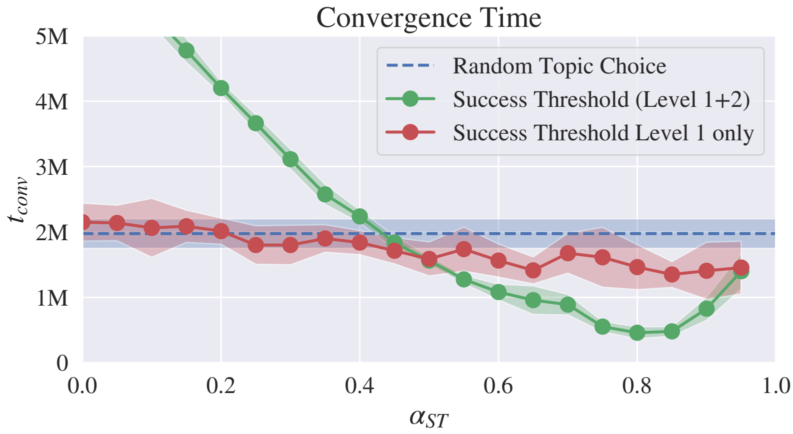


Figure 3.11: Convergence time dependence on parameter α_{ST}: a minimum is found around $\alpha_{ST} = 0.8$. The performance of Random Topic Choice is shown as reference. Restriction of the strategy to only first and second level are shown. $M = N = 100$, $W = \infty$, average over 8 trials.

Algorithm 9: **Minimal Counts strategy**, exploring when the minimum number of successes per meaning goes over a value n_{MC}.

Require: Success counts $S_C(m,t)$; parameter n_{MC}

procedure MinimalCountsTopicChoice($S_C(m,t)$; $m \in \mathfrak{M}_k$)
    if $\min\left(S_C(m,t)\right)_{m\in\mathfrak{M}_\mathfrak{k}} \leq n_{MC}$ **then** ▷ Level 1
        if $|\mathfrak{M}_\mathfrak{u}| > 0$ **then**
            Pick random m from $\mathfrak{M}_\mathfrak{u}$ ▷ Explore
        else
            Pick random m from $\mathfrak{M}$ ▷ Level 2.b
    else
        Pick $m = \arg\min_{m\in\mathfrak{M}_\mathfrak{k}}\left(S_C(m,t)\right)$ ▷ Level 2.a

---

parameter $\tilde{n}_{MC} = \frac{n_{MC}}{N}$ to be able to use the same value across different configurations with different values for N. A minimum can be found for $0.2 \leq \tilde{n}_{MC} \leq 0.7$, as can be seen on figure 3.12. On the same figure, we can see that both levels are needed to achieve significantly faster convergence than with Random Topic Choice, by comparing with convergence times associated with a modified strategy with only level 1. For further uses of this strategy, $\tilde{n}_{MC}$ will be set to 0.6, which ensures to stay close to minimum convergence time for different values of N ($N \in [10, 100, 1000]$, data not shown).

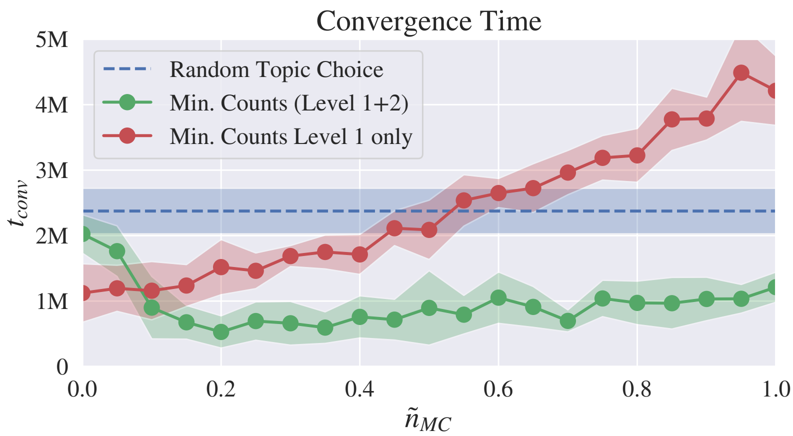


Figure 3.12: Convergence time dependence on parameter n_{MC}: a minimum can be found for $0.2N \leq n_{MC} \leq 0.7N$. The performance of the Random Topic Choice is shown as reference. $M = N = 100$, $W = \infty$, average over 8 trials.

3.4 *Global comparison*

In this section, we will compare three of the strategies to Random Topic Choice: Chunks, Success Threshold and Minimal Counts. Firstly in the normal setting, and then with homonymy and with a modified scenario.

3.4.1 *Scaling*

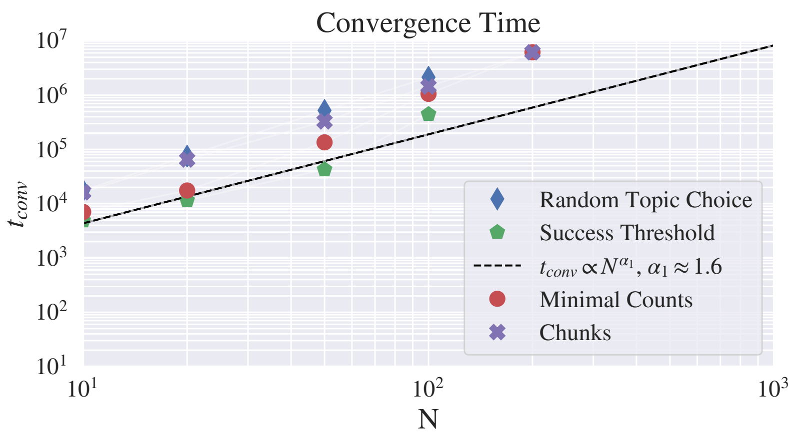


Figure 3.13: Comparison of different strategies under memory constraints. Scaling of t_{conv} with N. We can see that Success Threshold and Minimal Counts converge faster than the other strategies, for $N \leq M$. $M = 100$, averaged over 8 trials.

On figure 3.13, we can see that the Chunks strategy does in fact converge faster than Random Topic Choice, but not by much. Minimal Counts and Success Threshold do perform better, provided $N \leq M$. Over this value, the number of inventions is directly bigger than M, and agents fall quickly in the level 2.b (see description paragraph 3.1.3). Because the policy at this level is equivalent to random choice for both strategies, this behavior was to be expected.

3.4.2 *Homonymy*

When $W \approx M$, the same word can appear in several inventions, possibly for different meanings, creating homonymy. If $W = M$, the problem can become quite difficult, as there are less degrees of freedom, and convergence dynamics become extremely slow. We consider instead $W = 2M$, still low enough to quickly introduce homonymy.

On figure 3.14, we can see that the dynamics of Random Topic Choice become slower, and convergence time explodes when $N \geq 50$ (Simulations were stopped at 10^7 interactions). All other strategies keep their convergence properties.

3.4.3 *Hearer's Choice interaction scenario*

Memorization skills of infants are improved through active query of lexical knowledge Partridge et al., 2015, and experiments with children learning tasks in a social context suggest that this active behavior may also be part of the mechanisms used naturally in an

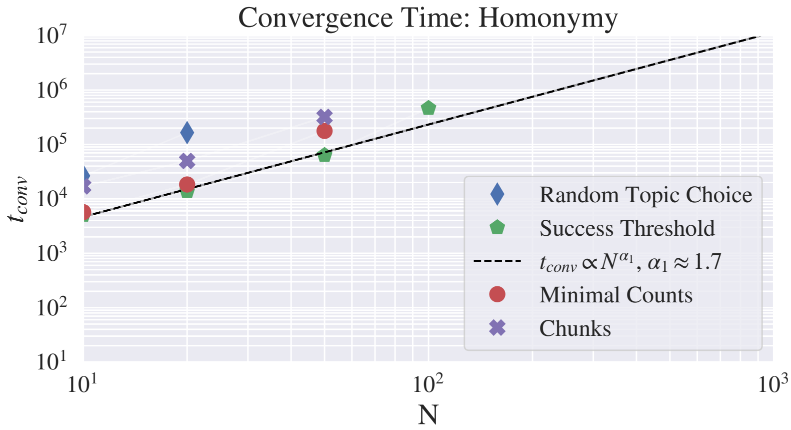


Figure 3.14: Comparison of different strategies under memory constraints, with homonymy. Scaling of t_{conv} with N. We can see that Success Threshold and Minimal Counts converge faster than the other strategies, for $N \leq M$. Chunks converges faster than RTC, whose value of t_{conv} exceeds 10^7 for $N \geq 50$ $M = 100$, averaged over 8 trials.

interacting population of human learners Vredenburgh and Kushnir, 2015. In our case, only the speaker has an active behavior, what if it would be the hearer?

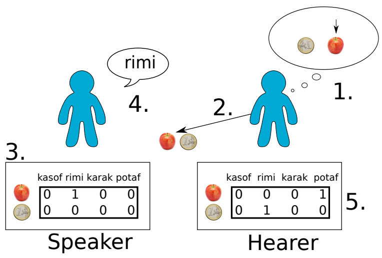


Figure 3.15: In Hearer's choice scenario, the first step changes: Hearer is choosing the topic instead of Speaker. See main text for a detailed description of the scenario.

Hearer's Choice is a modification of the interaction scenario, where the topic is chosen not by the speaker $\mathcal{S}$ but by the hearer $\mathcal{H}$. The exact scenario, illustrated in figure 3.15 and comparable to the standard scenario presented in section 2.1.4, is the following:

1. $\mathcal{H}$ chooses a topic m
2. $\mathcal{H}$ indicates the intended meaning m (by pointing at it for example)
3. $\mathcal{S}$ checks its vocabulary to find or invent a word w associated to m
4. $\mathcal{S}$ utters the word w
5. $\mathcal{H}$ guesses a meaning m_H from w using its vocabulary, and compares it to m.

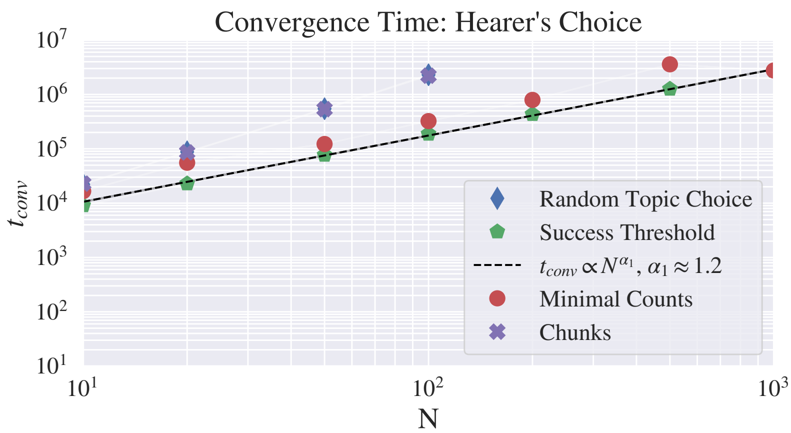


Figure 3.16: Comparison of different strategies under memory constraints, with the Hearer's Choice scenario. Scaling of t_{conv} with N. We can see that Success Threshold and Minimal Counts converge faster than the other strategies, even for $N \geq M$. $M = 100$, averaged over 8 trials.

On figure 3.16, we can see that it is possible with this modified scenario to perform better than Random Topic Choice, which is by definition the same as in the standard interaction scenario. Other strategies keep their properties for low values of N, but scale better with increasing N, and interestingly keep this behavior above the threshold $N \approx M$. The dynamics of Success Threshold in this case are similar to Random Topic Choice without memory constraints, even slightly faster.

3.5 Discussion

We have described Active Topic Choice in 2 levels: exploration or exploitation (level 1), and when exploiting, which meaning to choose amon all known ones (level 2). Level 2 can, be split into two subcases: there are still meanings to explore (level 2a), or there are no more (level 2b).

We have introduced a few strategies for the Active Topic Choice, under strong memory constraints. A first group of them (Information Gain maximization, Chunks) only rely on the size of the lexicon, which corresponds to the minimum memory usage M, for the lexicon itself. We have seen that those strategies can be described by a single vector of values in $[0;1]$ and of dimension $M+1$. For small values of M, optimization algorithms converged to a vector close to the Chunks strategy. This strategy converges a bit faster than Random Topic Choice or Information Gain maximization, but scales the same against N. The fact that Chunks is better than Information Gain maximization can be interpreted by the fact that it results in less inventions, and therefore less synonyms within the population.

A second class of algorithms (Success Threshold, Minimal Counts) are using the count of past successes and failures per meaning. In terms of memory, this has a cost of $2M$ (two values per meaning), leading to a total of $3M$ if counting the lexicon. For a population size lower than $2M$, they converge faster, and scale better than Random Topic Choice. The limit at $N = 2M$ is related to a direct shift towards level 2b of the strategy, because the number of inventions for the first interactions of agents ($\frac{N}{2}$) becomes greater than the number of meanings M, and there immediately no more meanings to explore.

It can be noted that level 2a for both strategies uses an *incompetence max* approach: choosing the option associated with the lowest competence. See Oudeyer and Kaplan, 2009 for a review of possible intrinsic rewards.

Level 2b is in this chapter defined as equivalent to random topic choice, so when the limit $N = 2M$ is reached, strategies tend to have dynamics more similar to random topic choice. This choice was done because an extrapolation of level 2a slows down the dynamics: this is probably caused by waves of reinforcement for the conventions that become less present in the population, until another convention becomes rarer, and gets reinforced on their turn (data not shown). The incomplete lexicon has more stability in this way, and getting out of this pseudo-equilibrium takes a high number of interactions.

However, if the alternative scenario Hearer's Choice is used, this can be overcome, because inventions for a meaning m propagate only if the hearer asked for this meaning. Agents can stay in level 2a (i.e.

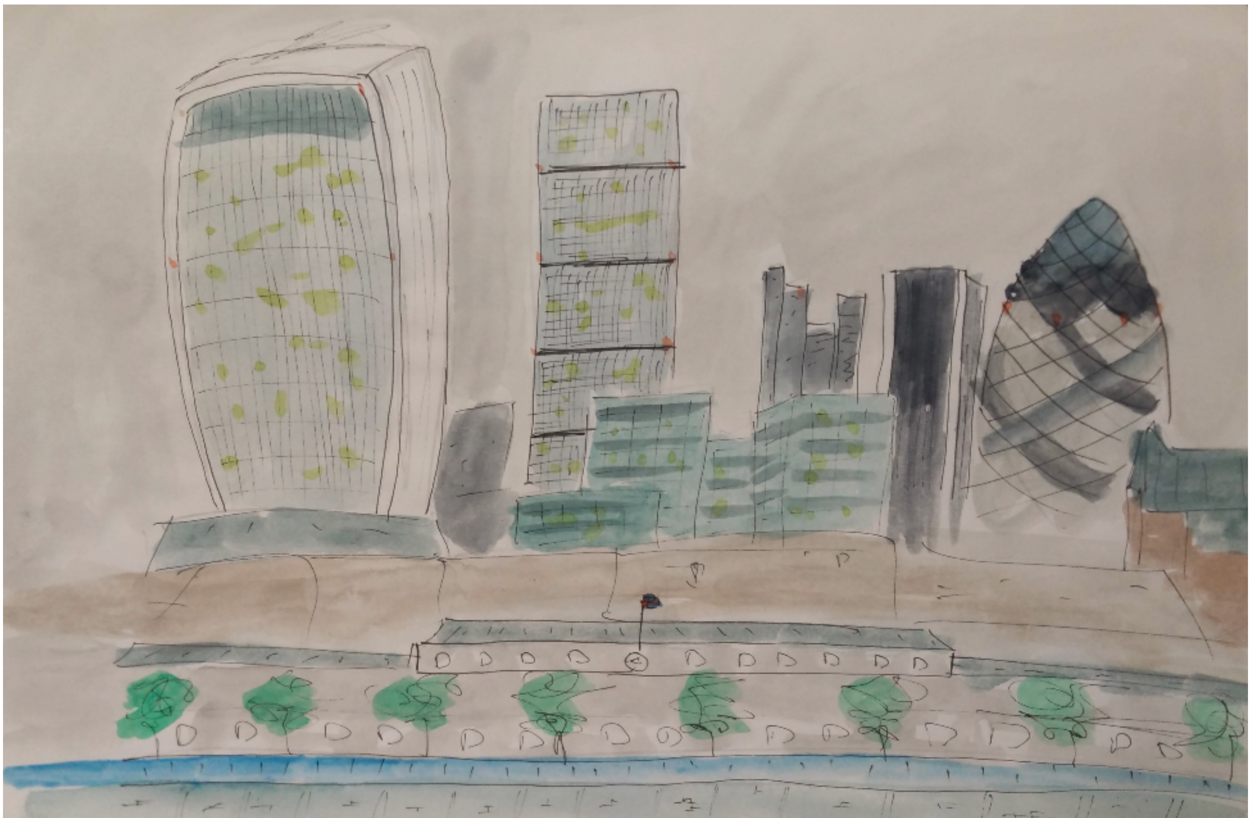
still having meanings to explore) even if $N \geq 2M$. Hearer's Choice is particularly efficient in this setting.

The strategies are robust to a bias towards the introduction of homonymy by limiting W: their dynamics stay the same for $N \leq M$. Random Topic Choice on the contrary sees its convergence time burst (data not even visible on figure 3.14 for $N \geq 50$).

It seems that for most strategies, even if we restrain memory by using the imitation vocabulary update, refraining from exploring is a key mechanism in improving the global dynamics. For example in 3.10, we can see the hierarchy Chunks > Explo. Biased > Info. Gain. This correlates with the respective biases towards exploration and the number of inventions being done.

Considering the strategies using more memory than the sole lexicon, Success Threshold and Min. Counts, they depend on a parameter, which we could optimize in our case, but this may be harder for different contexts, as the parameters are continuous and abstract. Searching for optimal values in the parameter space can be quite costly computationally, and if it is possible that they might be found through evolutionary processes (considering the actual behavior of people), this seems unlikely, especially if it depends on the context and population size for example. We will continue this part of the discussion (about abstract parameters) in the following chapter, which will start by reintroducing synonyms and homonyms in the strategies of this chapter.

Last but not least, it can be noted by a quick comparison of figures 2.7 and 3.13 that Success Threshold converges more or less as fast as Random Topic Choice without memory constraints. This means that we already have a candidate for our optimization problem: really low memory usage, but same convergence speed.



Neurocuriosity Conference, London. November 2016

4
Principled measures

Contents

What if we allow homonymy and synonymy? They both exist in real languages, and it is not inconcievable that in processes like the Naming Game with a drive towards one-to-one mappings, we maintain temporarily a few conflicting hypothesis before discarding some of them, while keeping the average memory low. In this context, can we still harness the complexity growth, while keeping similar dynamics? Or is it possible to converge even faster?

In this chapter, we will clarify what we are trying to optimize in the standard Naming Game (without forbidding homonymy and synonymy), and translate it as a measure locally available to agents: the Local Approximated Probablity of Success, or LAPS[1]. New Topic Choice strategies will be explicited, driven by a maximization of this measure. These strategies allow to converge faster than with Random Topic Choice, control efficiently local complexity growth, and do not necessitate a lot of memory.

[1] A table of symbols, acronyms and abbreviations can be found at the last page of this manuscript.

4.1 Some strategies from last chapter

Let us begin by directly reusing the strategies introduced at the last chapter. Only Information Gain maximization cannot be reused directly: the measure of information first needs to be extrapolated to the use of synonyms and homonyms (see next section).

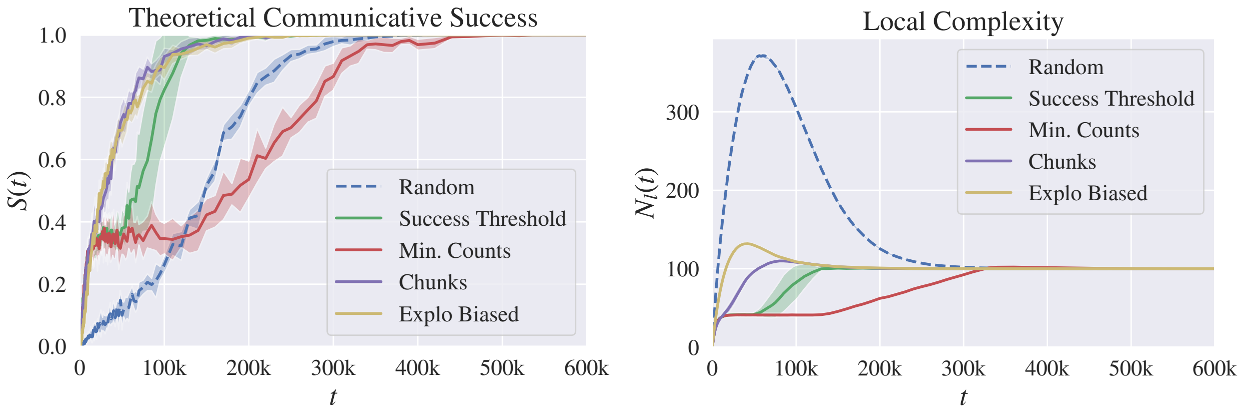


Figure 4.1: Allowing synonymy and homonymy for a few strategies of last chapter: Communicative success and local complexity. Globally, a low memory is conserved, and convergence is for some of them faster then Random Topic Choice. $N = 100$, $M = 100$, $W = \infty$, averaged over 8 trials.

We can see that the active mechanisms do help to naturally keep a low memory, however the optimal parameters are not the same anymore for Success Threshold and Minimal Counts, confidence is clearly underestimated as a stop in exploration is observed (plateau in local complexity). Chunks and ExploBiased are faster and do not exhibit such a plateau in complexity, but it is probably possible to do better since they do not use any information of past interactions. A first possibility would be to – again – study convergence dependence on the parameters used for Success Threshold and Minimal Counts. These parameters are continuous (if we consider the optimal parameter of MC as a ratio over M), and their optimal values seem arbitrary. Finding them could result in a parameter optimization, which can be quite fastidious[2]. Trying to determine them analytically seems even more fastidious, as they are not related to other values. Would there be other strategies with simpler parameters: possibly principled, and maybe even not continuous but discrete?

[2] N.B.: The authors spent a non-neglectable amount of time trying this approach.

4.2 What should an agent optimize?

4.2.1 Issue with entropy measures

From our omniscient point of view of simulator of the Naming Game, we can tell when the system has converged, or to which extent it has converged. Our main measure for this purpose is the *TCS* (Theoretical Communicative Success[3]) noted $S(t)$, described in paragraph 2.2.2 and used extensively in the previous two chapters. However, this measure is global, and not directly accessible to individual agents. They cannot know for sure when the system has converged, let alone estimate an intermediate *TCS* value.

[3] A table of symbols, acronyms and abbreviations can be found at the last page of this manuscript.

In the previous chapter, we have introduced a measure of information, and used it along a couple of assumptions about the state of other agents to define the Information Gain maximization strategy. Can we extend this value to the situation we are now studying, i.e. with synonymy and homonymy? Information measures are more complicated in this case, and would be expressed as a negative entropy, or *negentropy* of the lexicon. Several measures of entropy can be used:

Word entropy: The sum of the entropy of each vector corresponding to a meaning m:

$$-\sum_{m\in\mathfrak{M}_k}\sum_{w\in\mathfrak{W}_k(m)}\mathcal{V}^c_{mw}\log_2\left(\mathcal{V}^c_{mw}\right)=\sum_{m\in\mathfrak{M}_k}\log_2\left(|\mathfrak{W}_k(m)|\right) \quad (4.1)$$

Meaning entropy: The sum of the entropy of each vector corresponding to a word w:

$$-\sum_{w\in\mathfrak{W}_k}\sum_{m\in\mathfrak{M}_k(w)}\mathcal{V}^d_{mw}\log_2\left(\mathcal{V}^d_{mw}\right)=\sum_{w\in\mathfrak{W}_k}\log_2\left(|\mathfrak{M}_k(w)|\right) \quad (4.2)$$

Combination of the two: The sum of both.

Where $\mathfrak{W}_k(m)$ is the set of known words for meaning m, and $\mathfrak{M}_k(w)$ is the set of known meanings for word w.

We will consider the combination of both. We can note that there is a particularity with empty vectors (for either words or meanings): should they count as maximal entropy ($\log_2(|\mathfrak{M}|)$ and $\log_2(|\mathfrak{W}|)$), or minimal (0)? The formula as it is would be an empty sum, i.e. they would count as 0, as the corresponding values in the matrix are set to 0. But because no knowledge has been acquired yet for the corresponding meaning or word, they still carry a high ambiguity. Instead, we can re-use our previous measure of information on the subset of vocabularies without synonymy and homonymy, and make the negentropy measure its extrapolation to the set of all possible lexicons. Considering this last option, the final entropy measure is:

$$E(t)=\sum_{m\in\mathfrak{M}_k}\log_2\left(|\mathfrak{W}_k(m)|\right)+\sum_{w\in\mathfrak{W}_k}\log_2\left(|\mathfrak{M}_k(w)|\right)+\sum_{j=|\mathfrak{M}_k|}^{M-1}\log_2(W-j) \quad (4.3)$$

For our agents, we are seeking a functional to either maximize or minimize, would entropy suit this role? Even if the global evolution of the negentropy is globally decreasing to a minimum (see figure 4.2), we can spot two issues:

The number of possible states explodes: If we try to optimize the measure of the other agent in a similar way as in for the Information Gain maximization strategy, even determining the actual dimension of the equivalent of the previous decision vector is hard, but a rough lower bound would already be $2^{\frac{M(M-1)}{2}}$. The dimension is the number

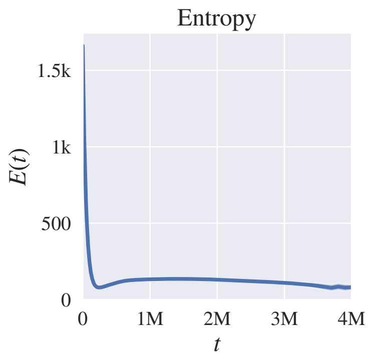


Figure 4.2: Typical evolution of the entropy measure: it is not monotic, but goes through a local minimum before reincreasing and finally reaching 0. Random Topic Choice, $N = 1000$, $M = 100$, $W = \infty$, averaged over 8 trials.

of distinct states of the system. States are distinct if they cannot be transformed into each other by any permutation of rows and/or columns or the lexicon. Consider a lexicon that is a triangular matrix – possibly after a permutation of rows and/or columns – with 1 everywhere on the diagonal. Each coordinate of the triangle can take an arbitrary value, all corresponding states would be distinct. The given lower bound is the count of these states; but of course there are many more possible states. Eventually, the important message is: this value is too high anyway.

Dynamics are locally non-monotonic: As an example, consider a given meaning m: the first interaction with m will yield a great decrease in entropy; as there will be only one word associated to it. Afterwards, it can only increase, before reaching again the lowest entropy state after a successful interaction. This particularity can be observed as well on a global scale (see figure 4.2). A direct minimization of the entropy would force the system too fast in this local minimum, which corresponds to a behavior heavily biased in favor of exploration at the beginning, creating many conflictual associations, which is what we would like to avoid (see chapter 5 for more information about the importance of the number of inventions).

To sum up, if we do not constrain the system to avoid homonyms and synonyms but at the same time do not allow usage of past interactions information, the best strategy so far seems to be the Exploration Biased topic choice (see algorithm 5). However, considering the information carried by past interactions, could we redefine another functional to optimize?

4.2.2 *Interactions as an information sampling process*

Let us recapitulate what is the information brought by a single interaction:

Success $b_{success}$: Boolean, success or failure of communication

Topic m_S: Meaning chosen by the speaker

Word w: Used by the speaker to refer to m_S

We can note that the identity of the other agent is absent: if the algorithms could consider it, intermediary languages could potentially emerge between pairs of individuals, instead of a shared lexicon at population level. It is thus difficult to infer the state of the hearer, because the speaker not only does not interact much with a given individual, but also because it is not known when an interaction happens with the same individual. In other words, beliefs can only be built on the *state of the population* and not on individual states. In fact, interactions can be seen as an *information sampling process* where beliefs are built about an average state of the population.

We introduced in paragraph 2.2.2 the notion of an average vocabulary of the population $\mathcal{V}(\mathfrak{P})$; which is exactly what we are seeking. Trying to coordinate with other agents narrows down to coordinate with this average population lexicon. This idea is not new, and several vocabulary update policies used in previous work reflect this idea[4]. However, this also results in a burst of lexicon complexity, which is still our main issue.

[4] e.g. Lateral Inhibition (Wellens, 2012) or trying to entirely copy at each interaction the guessed state of the population (Oliphant, 1999)

Could agents build a partial representation $\widetilde{\mathcal{V}}(\mathfrak{P})$ of the average vocabulary $\mathcal{V}(\mathfrak{P})$? Partial in the sense that it uses only the sparse information provided by the interactions, but also because it should not take too much memory (and should therefore stay distant from the actual state of the population, which has by definition a greater complexity – N_d – than the individual lexicon). To do so, we will limit memory by considering a sliding time window.

4.2.3 Reconstructing an average population lexicon

We construct independently the coding and decoding parts $\widetilde{\mathcal{V}}^c(\mathfrak{P})$ and $\widetilde{\mathcal{V}}^d(\mathfrak{P})$. For every meaning m (and every word w), we use a sliding window over the recent past interactions – of maximal length τ, the time scale parameter– and count the number of times it is associated to each word w' (or meaning m'). This value divided by τ is the local estimation of the probability of an other agent coding m using w' (or decoding w as m'). With this, we retrieve the values of both matrices $\widetilde{\mathcal{V}}^c(\mathfrak{P})$ and $\widetilde{\mathcal{V}}^d(\mathfrak{P})$.

Let $M^c(m)$ be the memory of the past interactions where m was the topic, if there has been T_m such interactions. w_t denotes the word used during the t^{th} interaction of the agent using the meaning m. We can now build $\widetilde{\mathcal{V}}^c(\mathfrak{P})$:

$$M^c(m) = (w_t)_{1 \leq t \leq T_m} \qquad \widetilde{\mathcal{V}}^c(\mathfrak{P})_{mw} = \frac{\sum\limits_{t=T_m-\tau+1}^{T_m} \delta_{w,w_t}}{\tau} \tag{4.4}$$

Similarly, by defining $M^d(w)$ be the memory of the past interactions where w was the topic, with T_w such interactions, we can build $\widetilde{\mathcal{V}}^d(\mathfrak{P})$:

$$M^d(w) = (m_t)_{1 \leq t \leq T_w} \qquad \widetilde{\mathcal{V}}^d(\mathfrak{P})_{mw} = \frac{\sum\limits_{t=T_w-\tau+1}^{T_w} \delta_{m,m_t}}{\tau} \tag{4.5}$$

Until τ interactions have been done with a given meaning or word, $\sum\limits_{w} \widetilde{\mathcal{V}}^c(\mathfrak{P})_{mw}$ and $\sum\limits_{m} \widetilde{\mathcal{V}}^d(\mathfrak{P})_{mw}$ do not sum to 1. The remaining probability weight is assumed to be associated with failure. If we would normalize to 1, with a single interaction an agent would already estimate as 100% sure that the same word-meaning association would be used again with the same topic for example. Without the normalization, this happens only after τ interactions. In other words, this reflects lack of information due to small sample size.

The memory needed to build this average vocabulary scales with $M \cdot \tau$. If τ stays constant and small, this means that it scales with M, size of the completed lexicon. When using this type of memory, if we manage to control the local complexity growth, we will still have to check that τ keeps a small value to avoid simply shifting the complexity growth to another type of memory.

4.2.4 *LAPS measure*

Now that we have built a representation of the average state of the population, how should we use it? Simply by using the same measure that we use at a global level to characterize convergence, the TCS. We define the Local Approximated Probability of Success, a local equivalent of the Theoretical Communicative Success for an agent with vocabulary $\mathcal{A}$ having built a representation of the population vocabulary $\widetilde{\mathcal{V}}(\mathfrak{P})_{\mathcal{A}}$:

$$LAPS_{\mathcal{A}} = TCS\left(\mathcal{A}, \widetilde{\mathcal{V}}(\mathfrak{P})_{\mathcal{A}}\right) \tag{4.6}$$

For simplicity, and to express its similarity to $S(t)$ it will be noted $S_{LAPS}(t)$. The $LAPS$ measure is a suitable functional to maximize:

Monotonicity: The measure is globally monotonic (see figure 4.3). Locally, it does not exhibit the properties of the negentropy: it does not need to go through a maximum before decreasing and augmenting again. A strategy based on its maximization is possible.

Parameter: The measure depends on one single parameter, the time scale τ, which is both principled (it is a direct measure of the extent in the past of the memory of past interactions), and discrete: a potential search in parameter space for optimization will be easier. Moreover, the suitable values for the parameter should be low, because a high parameter also means higher memory usage.

Measure of confidence: It can directly be seen as a measure of confidence, which can reach 100%. No need to add a threshold like in previous strategies for designing the first level of a strategy.

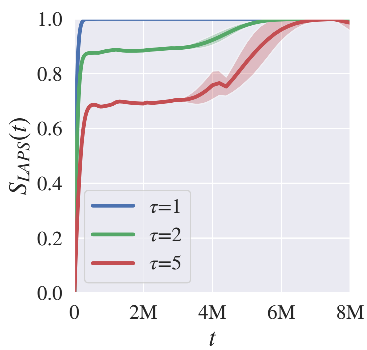


Figure 4.3: Typical evolution of the $LAPS$ measure. $N = 1000$, $M = 100$, $W = \infty$, averaged over 8 trials.

4.3 *Derived strategies*

4.3.1 *Exact value*

An intuitive approach to the maximization is a greedy algorithm (for a topic choice policy), maximizing the variation of $LAPS$ at each interaction. In other words, the speaker should choose the meaning m yielding the maximum expected ΔS_{LAPS}, presented in pseudo-algorithms 10 and 11.

Algorithm 10: **Computing ΔS_{LAPS}**

Require: Word w, Meaning m, Success $b_{success}$, Lexicons $\mathcal{V}$, $\widetilde{\mathcal{V}}$

procedure $\Delta S_{LAPS}(m, w, b_{success}, \mathcal{V}, \widetilde{\mathcal{V}})$

$\quad \mathcal{V}_{new} = \text{Update}(\mathcal{V}; m, w, b_{success})$

$\quad \widetilde{\mathcal{V}}_{new} = \text{Update}\left(\widetilde{\mathcal{V}}; m, w, b_{success}\right)$

$\quad$ **return** $TCS(\mathcal{V}_{new}, \widetilde{\mathcal{V}}_{new}) - TCS(\mathcal{V}, \widetilde{\mathcal{V}})$

Algorithm 11: **LAPSmax strategy**, maximizing the expected increase of the LAPS measure. Level 2b is not described: all meanings are equivalent for the agent in the corresponding case.

Require: Success ratios $S_R(t)$ and $S_R(m,t)$; parameter α_{ST}

```
procedure LAPSmaxExactTopicChoice(S_R(m,t); m ∈ 𝔐_k)
    if S_LAPS(t) = |𝔐_k|/M then                          ▷ Level 1
        Pick random m from 𝔐_u                           ▷ Explore
    else
        Pick m = arg max_{m∈𝔐_k} (E(ΔS^{LAPS}_{t→t+1}))   ▷ Level 2
```

Evaluating the argmax is quite costly computationally: one would need to evaluate the resulting $LAPS$ for each tested m, w. Per agent, this is over $N_l(t)$ times, considering possible inventions. An alternative is to consider a Monte Carlo estimation, but the number of evaluations would still be high. As the $LAPS$ measure has to be computed over the whole lexicon, we can only afford one or two evaluations per interaction. Above this, it becomes cognitively implausible.

4.3.2 *Multi-Armed Bandits*

We can however compute the $LAPS$ value at each interaction, and see the system as a black box, outputting ΔS_{LAPS} after each interaction. As presented in paragraph 3.1.1, Multi-Armed Bandits(Bubeck, Cesa-Bianchi, et al., 2012) are a set of algorithms that can help solve this problem: following a decision between a finite set of options, a reward value is obtained and used to update the choice policy. The name Multi-Armed Bandit comes from an analogy with a person trying to maximize their gain while facing a set of slot-machines (also called *one armed bandit*), and being able to use only one at a time. The probability distribution of the reward of each machine is unknown, and the player has to both collect information by playing and exploit the highest rewarding machine – with limited knowledge of its reward distribution – hence keep balance between exploration and exploitation. In our problem, we can see known meanings as the possible arms, and the reward ΔS_{LAPS}.

Our case is quite specific, as:

1. Reward distributions are non stationary
2. Reward distributions depend on past choices
3. The number of arms grows over time (and starts at 0).

This situation led us to choose an algorithm where weights associated to each arm undergo a decay over time, which let them stay at the same order of magnitude of the initial weights of new arms. We took inspiration from Clement et al., 2015, where a similar algorithm is used to model young students learning math.

The resulting algorithm depends on two parameters: integrated balance between reward-driven exploitation and random exploration between arms through the parameter γ [5], and time scale n for the decay of weights. As a reward, we consider the increase of LAPS

[5] to avoid being stuck and ensure that all meanings have a non-zero probability to be chosen

yielded by the interaction, ΔS_{LAPS}, or 0 if the latter is negative in order to avoid negative weights. See algorithm 12.

Algorithm 12: **LAPSmax bandit**, estimated argmax for the LAPSmax strategy using a Multi-Armed Bandit. At the end of an interaction with a new meaning, a new arm is created with a weight w_a equal to the reward r_i obtained. Level 2b is not described: all meanings are equivalent for the agent in the corresponding case.

Require: parameters γ rate of exploration for bandit
Require: n time scale for weights decay

procedure LAPSmaxTopicChoice($S_R(m,t)$; $m \in \mathfrak{M}_k$)
  if $S_{LAPS}(t) = \frac{|\mathfrak{M}_k|}{M}$ **then**
    Sample m from $\mathfrak{M}_u$
  else
    for $a \in Arms$ **do**
      $\tilde{w}_a = \frac{w_a}{\sum_j w_j}$
      $p_a = (1-\gamma) \cdot \tilde{w}_a + \frac{\gamma}{|\mathfrak{M}_k|}$
    Sample $m \in Arms$ using distribution $(p_a)_{a \in Arms}$
  return m

  {Interact using topic m and compute reward r}

  if $m \in Arms$ **then**
    $w_m \leftarrow \frac{n}{n+1} \cdot w_m + r$
  else
    Add m to $Arms$ with $w_m = r$

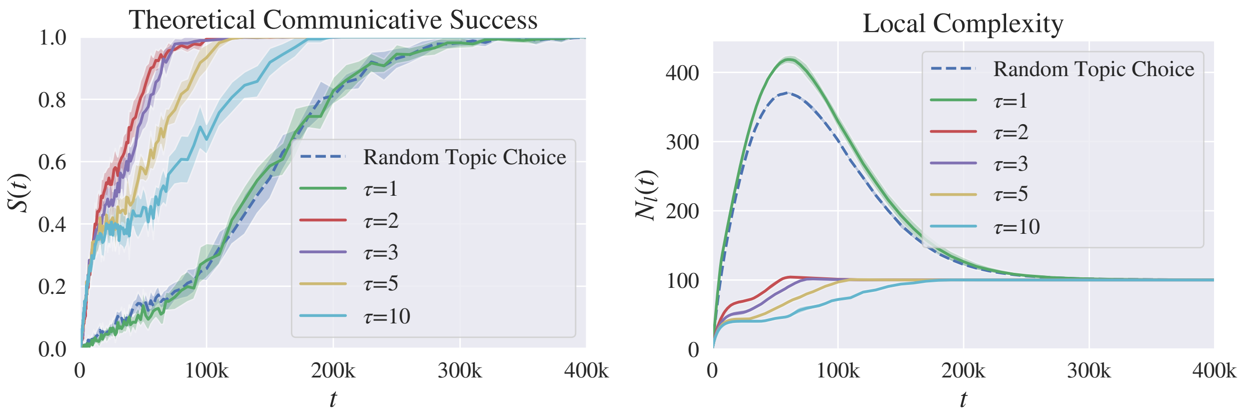


Figure 4.4: Evolution of the system following the LAPSmax with Bandit strategy, for $S(t)$ and $N_l(t)$. If τ increases, agents wait longer to be confident to explore. $M = 100$, $N = 100$, $W = \infty$, averaged over 8 trials.

The LAPSmax strategy converges faster than any other strategy seen so far, and keeps memory usage under the final size of the lexicon M, for $\tau = 2$ and $\tau = 3$ in the shown example figure 4.4. The minimum $\tau = 2$ is stable across different values of N (from 10 to 1000, data not shown), corresponds by definition to a lower memory usage than $\tau = 3$, and will be considered as the standard value for this strategy in the remainder of this thesis. The case $\tau = 1$ is an outlier, being a simple autocorrelation with the current interaction, and not a real comparison with data collected after past interactions. For values of τ greater than 2, agents need more time to get confident about their lexicon, and dynamics are slowed down. (See figures 4.4 and 4.5)

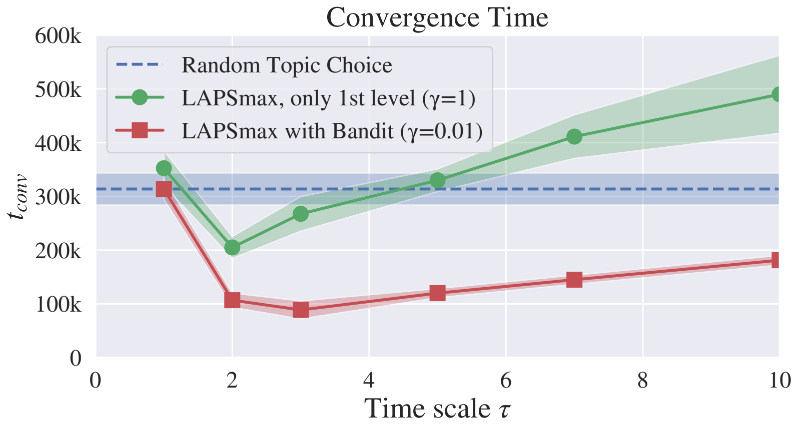


Figure 4.5: Dependence of the LAPSmax strategy with time scale parameter τ. Optimal convergence time is attained at $\tau = 2$ and $\tau = 3$. $\tau = 1$ is particular, see main text. Both levels of the strategy contribute to faster convergence. $M = 100$, $N = 100$, $W = \infty$, averaged over 8 trials.

4.3.3 *Coherence Strategy*

The previous strategy works quite well, but is a bit costly computationally, heavily depends on the state of the whole lexicon, and a part of it works as a blackbox. We can simplify the $LAPS$ measure by considering only past information about meanings (not taking homonymy into account), and because we are sure that the last used word w (for meaning m) is in the lexicon, consider only this one. The $LAPS$ measure is then proportional to the number of occurences of w in the stored memory of past interactions for meaning m. We call this last measure the *Coherence* measure $C(m,t)$. By definition, its value is an integer between 0 and τ. As for Success Threshold and Minimal Counts, we can design a strategy following an *incompetence max*[6] approach for level 2 (see algorithm 13).

[6] Choosing the option associated with the lowest competence. See Oudeyer and Kaplan, 2009 for a review of possible intrinsic rewards.

procedure COHERENCETOPICCHOICE
 if $C(m,t) = \tau;\ \forall m \in \mathfrak{M}_k$ **then** ▷ Level 1
  Pick random m from $\mathfrak{M}_u$ ▷ Explore
 else
  Pick $m = \arg\min_{m \in \mathfrak{M}_k} (C(m,t))$ ▷ Level 2

Algorithm 13: **Coherence strategy**, choosing known meanings with the current minimum coherence. Level 2b is not described: all meanings are equivalent for the agent in the corresponding case.

Figure 4.6 shows us that the Coherence strategy can also converge faster than Random Topic Choice, but even faster than LAPSmax, and both levels contribute as well to this property. If the dependence on parameter τ resembles the one for LAPSmax [7], the optimal value is not as simple as for LAPSmax, with the settings of the experiments presented in the figure it would be $\tau = 8$.

[7] i.e. starting above RTC for $\tau = 1$, going through a minimum, and then slowly increasing again

If we look at the data for other values of N, we can see that the optimum shifts towards greater values of τ. For a greater population, an agent needs to gather more information to be confident and continue exploring, which can explain this phenomenon. A discussion on a plausible expression for the optimal value for τ will be found in the next chapter, in paragraph 5.2.3, but we can already use the associated expression: $\tau_{opt} = \log_2(N) + 1$. Values obtained for $M = 100$ and $10 \leq N \leq 1000$ follow this tendency: the convergence time as-

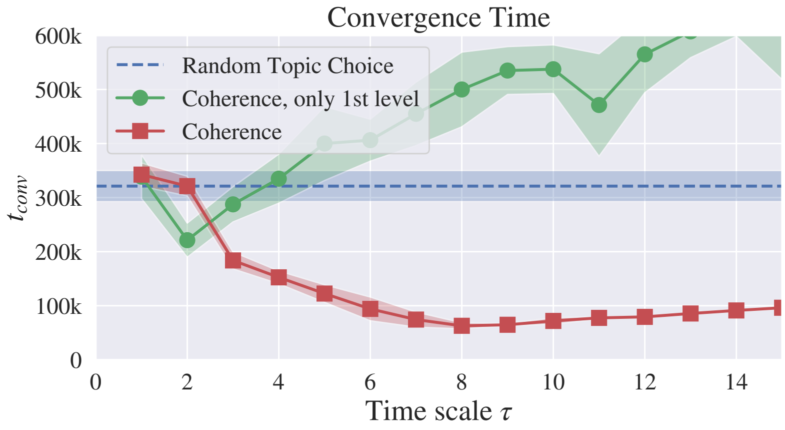


Figure 4.6: Dependence of the Coherence strategy with time scale parameter τ. Optimal convergence time is attained at $\tau \approx 8$. Both levels of the strategy contribute to faster convergence. $M = 100$, $N = 100$, $W = \infty$, averaged over 8 trials.

sociated to τ_{opt} is always close to the observed minimum (data not shown). A more thorough study of τ_{opt} would probably be useful to prove or refine this expression, but we will keep this expression as is in this thesis.

4.4 *Results*

4.4.1 *Scaling*

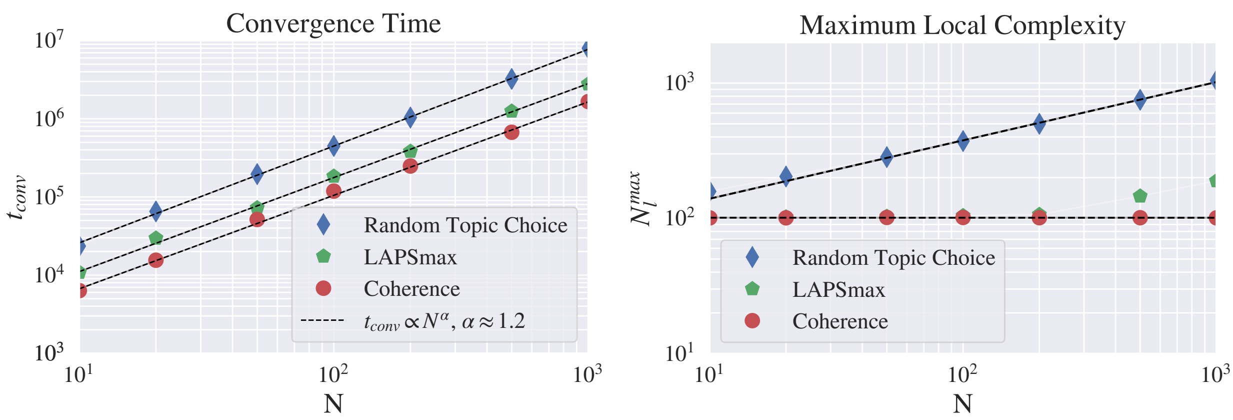


Figure 4.7: Scaling of LAPSmax and Coherence convergence time and maximum local complexity with N. They both scale more or less like RTC, but converge significantly faster. Memory usage is minimal (M), only for LAPSmax it increases a bit for high values of N. $M = 100$, $W = \infty$, averaged over 8 trials.

We can see on figure 4.7 that faster convergence is kept across various values for N for both strategies, and scales more or less like RTC. We will introduce other tools to better quantify the difference between these behaviors in the next chapter, developping a more theoretical approach. Memory usage is kept at the minimum value M, but for high values of N LAPSmax starts to need a bit more memory.

4.4.2 *Homonymy*

Both strategies are robust to the introduction of homonymy: the dynamics are quite similar for the different studied ratios between M

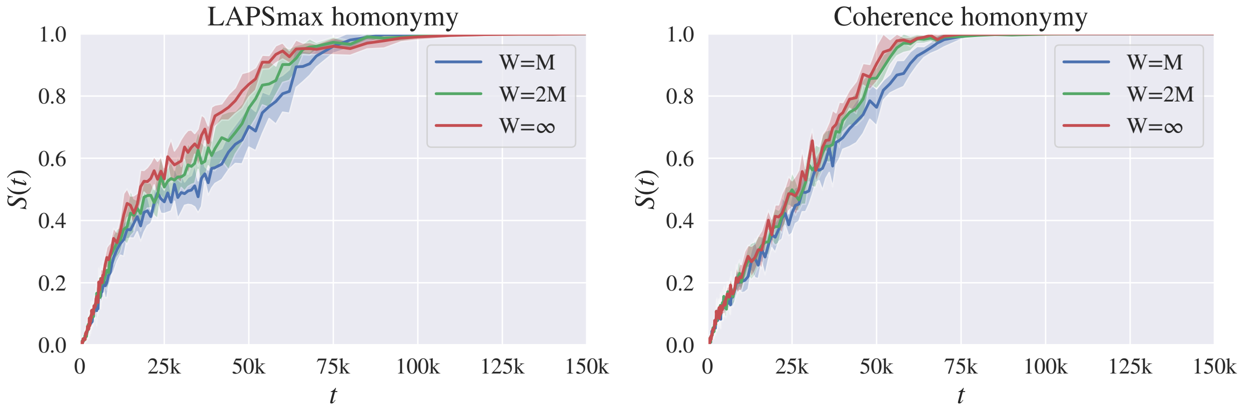


Figure 4.8: Impact of homonymy on LAPSmax and Coherence strategies, for different ratios between W and M. Both strategies are robust to these changes. $N = 100$, $M = 100$, averaged over 8 trials.

and W (see figure 4.8). Three phases can be distinguished: the first wave of invention, propagation, and agreement on the last meanings which had the most words. Slight differences can be observed in the second and final phases[8] of the convergence process, but compensate to reach the same values of convergence time.

[8] First phase: $S(t) > 40\%$, Second phase: $40\% \leq S(t) < 95\%$, Final phase: $S(t) \geq 95\%$

4.4.3 *Hearer's Choice*

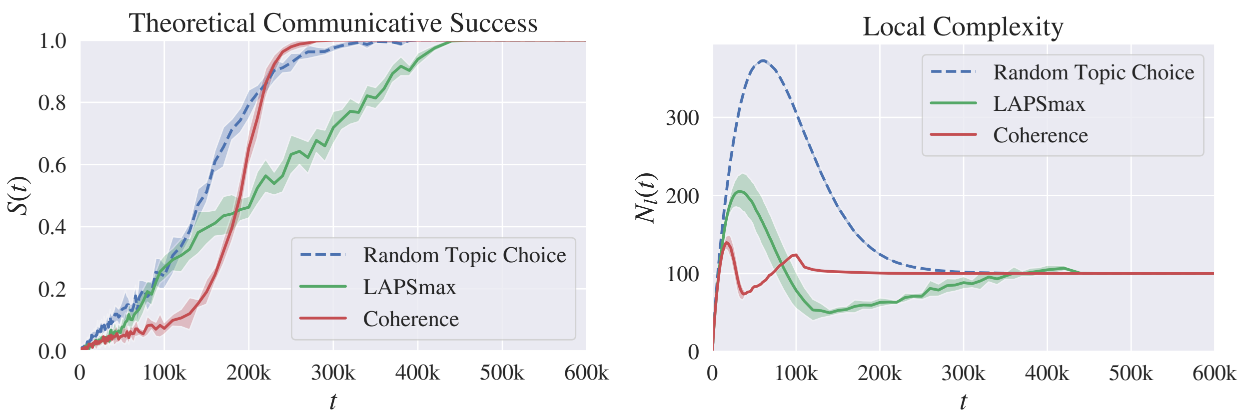


Figure 4.9: Hearer's Choice with LAPSmax and Coherence strategies. Hearer's Choice forces a first wave of many inventions, which slows down the convergence. $N = 100$, $M = 100$, $\tau = 5$ for LAPSmax, $\tau = 2$ for Coherence, averaged over 8 trials.

Hearer's Choice, when used with homonyms and synonyms, is problematic: for an agent at its second or third interaction, exploiting may still lead to an invention, if the speaker does not have any word for the chosen meaning. This results in a longer and higher wave of inventions, and creating a complexity burst, even if smaller than for RTC (see figure 4.9). We can still note that the end of the convergence process does not show a separate slower phase, as can observed for most of the strategies with Speaker's Choice (see for example figure 4.8).

4.5 Discussion

In this chapter, we have reintroduced the use of homonymy and synonymy, by using Minimal NG as a vocabulary update. We have seen that the strategies from last chapter can still converge faster than Random Topic Choice when adapted to this vocabulary update. However, abstract parameters have to be further tuned to get to this behavior, otherwise the dynamics can be slower than RTC.

We have introduced a new measure, *LAPS*, that measures locally the agreement of an agent with the population, and relies on a time scale parameter τ. This time scale controls the size of the recent memory used to compute the *LAPS* measure. We have introduced two strategies based on the maximization of this measure: LAPSmax and Coherence. LAPSmax is based on the exact expression of the *LAPS* measure, but uses a blackbox algorithm for the maximization, as the computation of its exact expected increase is computationally costly. Coherence is a simplified expression of the *LAPS* measure, but we can follow a heuristic based directly on its maximization. For LAPSmax, the behavior is similar to a *competence progress* drive, as for Coherence it is *incompetence max*, using the classification of intrinsic rewards found in Oudeyer and Kaplan, 2009.

The time scale parameter τ is easier to optimize than previous parameters: it is principled, discrete, easy to optimize [9], and we consider a bias towards small values to limit memory usage. It can be noted that with $\tau = 1$, the *LAPS* measure is only an auto-correlation with the current state of the agent, and does not really take into account the past. The value $\tau = 2$ is then the lowest possible value taking into account past interactions, taking the lowest amount of memory (compared to higher values of the time scale), and is credible for humans. The actual memory usage associated to the *LAPS* measure would be $\approx 2\tau M$, as values for both meanings and words are memorized. The optimal values for LAPSmax and Coherence being respectively 2 and $\log_2(N) + 1$, we can express the total memory usage associated to those strategies, including the final lexicon: $5M$ for LAPSmax, $M \cdot (2\log_2(N) + 3)$. The optimal value for LAPSmax may be underevaluated, and would need to also grow with N. However for $N \leq 10M$, it is still enough. For both strategies, the dependence on N and M for the optimal τ could still be studied further, but the given values are more than suitable for our range of parameters.

[9] The function to optimize, the convergence time, has a gentle slope leading to the minimum, while still staying at low values.

Both strategies are cognitively plausible, and converge faster than any other strategy seen so far, with a slight advantage for Coherence. This small difference is probably due to the fact that the blackbox algorithm used for LAPSmax introduces a delay through the evaluation of the weights associated to each meaning. While LAPSmax is more efficient memory-wise, Coherence is easier to compute. Depending on the context, one or the other could be more adapted to solve the problem from an algorithmical point of view.

If both strategies are robust against constraining to low values of W (and therefore introducing the necessity of regularizing homonymy), the slight differences observed (see figure 4.8) reveal an interesting separation in three distinct phases of the dynamics of the agreement process: inventions, propagation, and final discarding of competing associations. This mirrors the three phases of the dynamics of *adoption of novel variants* described in Fagyal et al., 2010: Innovation, Propagation/Selection, and Fixation. If the first two phases are well optimized for both strategies, the Fixation phase seems to slow down the whole process. The same remark applies for the strategies from the previous chapter, as seen in figure 4.1.

Interestingly, the Fixation phase does not suffer from this drawback when using the Hearer's Choice scenario. But this scenario, in this case, sees an unevitable burst of complexity, although smaller than with Random Topic Choice. This is due to the fact that *exploiting* as hearer can still result in an invention being made by the speaker. Hearer's Choice can therefore not be considered as a serious option.

Coherence and LAPSmax seem to have great properties, but would it still be possible to find even better performing strategies? What is the limit? The next chapter will introduce a theoretical analysis to try to answer this question.

Rome, January-July 2017

5
Theoretical approach

Contents

We have seen in paragraph 2.2 that in the standard Naming Game, agents typically go through a phase of high complexity, in terms of numerous synonyms or homonyms to remember. Only after reaching a peak do many of them start to be discarded and the agents eventually agree on a functional lexicon. In this chapter, we will first explicit analytical expressions of the characteristics of this peak, and link it to the number of inventions. We will then determine a lower bound to the number of inventions, and use this as a lower bound to convergence time when using active topic choice. Finally, we will use this lower bound to design performance measures and use them to compare the algorithms proposed in the previous chapters.

5.1 The memory peak

There are two complexity peaks in the typical evolution of the Naming Game: a global complexity peak, reached just after the last invention of a convention (see figure 5.2), and a local complexity peak, when all inventions have spread to the population (see figure 5.1). Complexity measures were defined in paragraph 2.2, and refer to the number of associations present in the lexicon of individual agents for the local complexity, and the number of associations present in the lexicon of at least one agent for global complexity. For the demonstrations presented in this section, we will consider $M = 1$ and $W = \infty$, unless specified.

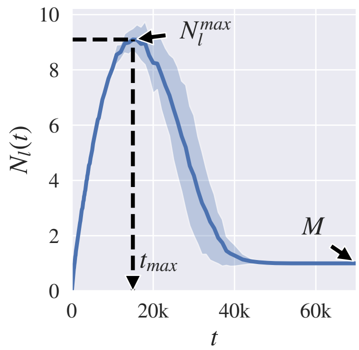


Figure 5.1: Typical evolution of local complexity. $M = 1$, $N = 1000$, averaged over 8 trials.

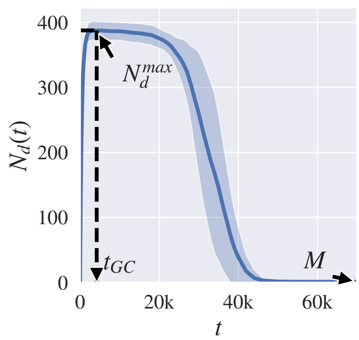


Figure 5.2: Typical evolution of the number of conventions present in the population, or global complexity N_d. $M = 1$, $N = 1000$, averaged over 8 trials.

5.1.1 Global Peak

The global peak is reached when all agents have at least one word for each meaning. In other words, that every agent has interacted at least once about every meaning, either as speaker or as hearer. Inventions happen when those first interactions are done as speaker. Because when using random topic choice there is a $\frac{1}{2}$ chance to be a speaker during this interaction, there is in total an expected value of $\frac{N \cdot M}{2}$ inventions, which is by definition the maximum reached at the peak of global complexity. The time needed to reach the peak is simply the time needed for every agent to talk at least once about every meaning. Let's consider the case $M = 1$. If we denote as u_t the expected number of agents having interacted at time t, and considering the probabilities that speaker and/or hearer of the next interactions have never interacted before[1], we have the following relation:

$$u_{t+1} = u_t + 2 \cdot \left(\frac{N - u_t}{N} \right) \tag{5.1}$$

Noting that $u_0 = 0$, we can find the solution:

$$u_t = N \cdot \left(1 - \left(\frac{N-2}{N} \right)^t \right) \tag{5.2}$$

The estimated time of the peak t_{GP} corresponds to the first value satisfying $N - u_t \leq 1$, i.e. $t_{GP} = \frac{\ln(N)}{\ln(N) - \ln(N-2)}$. When $N \gg 1$, $\ln(N) - \ln(N-2) \approx \frac{2}{N}$ and the expression becomes:

$$t_{GP} \approx \frac{N \cdot \ln(N)}{2} \tag{5.3}$$

5.1.2 Local Peak

The local peak is reached when conventions have spread through the whole population, and agents start to discard some of them. It was shown in Baronchelli, 2006 that the time to reach it follows a

[1] To avoid quadratic terms, we do not take into account the fact that hearer is chosen dependently of speaker, i.e. that it cannot be the speaker.

powerlaw of exponent $\frac{3}{2}$, and the maximum value a powerlaw of exponent $\frac{1}{2}$. By a novel approach, we will retrieve these results as well as find the proportionality constants associated to the powerlaws.

Let's consider the spreading process of a word w, and the number of agents having encountered it at least once u_t. Each time the word is used, u_t increases by 1 if the hearer is one of the $N - u_t$ agents who did not see it before. In the first phase of the Naming Game, we can assume that all conventions spread at a similar pace, and therefore all conventions (or words when $M = 1$) are used with an equivalent probability $\frac{1}{N_{inv}}$. As we have seen in the previous paragraph, $N_{inv} = \frac{N \cdot M}{2}$. With $M = 1$, that gives:

$$u_{t+1} = u_t + \frac{1}{N_{inv}} \left(\frac{N - u_t}{N} \right) = u_t + 2\frac{N - u_t}{N^2} \tag{5.4}$$

With solution:

$$u_t = N \cdot \left(1 - \left(1 - \frac{2}{N^2} \right)^t \right) \tag{5.5}$$

Let us now consider the evolution of $N_l(t)$, the local complexity, depending on the probability of success $p(S)$ and the probability that the hearer does not have the convention in its lexicon already:

$$N_l(t+1) = N_l(t) + (1 - p(S)) - p(S) \cdot (N_l(t) - 1) \tag{5.6}$$

We can note that when deletions are still rare (i.e. in the first phase of complexity growth) $N_l(t) = \frac{N_{inv} \cdot u_t}{N} = \frac{u_t}{2}$ and also $p(S) = \frac{u_t}{N}$. Using this and the previous equation, we get:

$$\Delta N_l(t) = \frac{N - u_t}{N} - \frac{u_t}{N} \left(\frac{u_t}{2} - 1 \right) \approx \frac{N - u_t}{N} - \frac{u_t^2}{2N} \tag{5.7}$$

The peak is reached when $\Delta N_l(t_{max}) = 0$, which translates to $u_{t_{max}} = \sqrt{2N+1} - 1 \approx \sqrt{2N}$ and finally, using equation 5.5:

$$t_{max} = \frac{\ln\left(1 - \sqrt{\frac{2}{N}}\right)}{\ln(N^2 - 2) - \ln(N^2)} \approx \frac{1}{\sqrt{2}} N^{\frac{3}{2}} \tag{5.8}$$

As for the value of the peak, we have it from $u_{t_{max}}$:

$$N_l^{max} = \frac{u_{t_{max}}}{2} \approx \frac{1}{\sqrt{2}} N^{\frac{1}{2}} \tag{5.9}$$

5.1.3 *Link with number of inventions*

If we repeat all the preceding without replacing N_{inv}, we get:

$$t_{max} \approx N \sqrt{N_{inv}} \tag{5.10}$$

$$N_l^{max} \approx \frac{N}{2\sqrt{N_{inv}}} \tag{5.11}$$

Seeing this, an intuitive idea to reduce the pick height and the corresponding time is to influence the value of N_{inv}. Luckily, it is exactly what is happening when using active topic choice: by exploring less, agents create less conventions, and N_{inv} gets smaller; though we will not be able to use the two equations above. In fact, we derived them from the usual dynamics of the Naming Game with random topic choice, and therefore they do not stand anymore. In the next section, we will adopt a new approach centered on N_{inv} to get information about t_{conv}.

5.2 A statistical lower bound to convergence

With active topic choice, agents restrain from inventing new conventions and stick to already existing ones (i.e. exploit, in the terms of chapter 3), in an extent depending on their strategy. In an ideal but highly improbable case, there would be only one invented convention per meaning. In other words, there are exactly M inventions happening during the corresponding simulation, one per each meaning.

After a short summary of what is known about the convergence time in the classical situation, with random topic choice, we will study the mentioned optimal case, which will give us a lower bound of the convergence time. Then, we will estimate the actual number of inventions and derive a new lower bound taking all this into account.

An important remark: the lower bounds have to be interpreted in a statistical way. For example, there is always a possibility that the system converges in really few steps ($\approx NM$) if the agents involved in each interaction happen to be picked in a specific order; but this situation is extremely unlikely. The lower bounds considered here are lower bounds to the statistical mean, not to the convergence time of a specific instance of the Naming Game.

5.2.1 Convergence Time with Random Topic Choice

The convergence time t_{conv} has been introduced in chapter 2. We call $t_{RTC}(N)$ the convergence time for a population of N agents talking about a single meaning. Its dependence on size of the population N was extensively studied in Baronchelli, 2006. It is a powerlaw of parameter $\frac{3}{2}$, as t_{max}. In fact, we have necessarily $t_{conv} \geq t_{max}$, hence the exponent of the powerlaw cannot be less than $\frac{3}{2}$. However, there are some log-periodic oscillations, modifying consequently the perceived exponent of the powerlaw if the span of values for N is not big enough, especially in the range $10^2 \leq N \leq 10^5$. A relatively accurate fit of the data, found in Baronchelli, 2006, gives us the formula:

$$t_{RTC}(N) \approx (2.3 + \sin(1 + 0.4\ln(N))) \cdot N^{\frac{3}{2}} \tag{5.12}$$

In the following paragraphs, we will use the values of $t_{RTC}(N)$. In practice, those are recomputed for small values ($N \leq 100$, average over 20 iterations), and the formula used for values $N > 100$. Concerning the precision of the fitted constants, they will not interfere at all in this work, as we will not study $N > 100M$.

5.2.2 *Convergence time with single invention per meaning*

We will here consider the case where there is only a single invention per meaning. We will focus on the dynamics of only one meaning m, and redefine the time t as the number of interactions having involved m only. Let u_t be the number of agents having adopted or been exposed to the meaning m. They all use the same word w, as there has been only one invented convention with meaning m. Under our assumptions, new interactions involving m have necessarily as speaker an agent among the u_t ones having been exposed to the convention. Increase of u_t will only happen if the hearer is not among them:

$$u_{t+1} = u_t + \frac{N - u_t}{N} \tag{5.13}$$

We set $u_0 = 0$. A more exact solution would be to consider $u_1 = 2$ as an inital condition, as the first interaction spreads the convention to both hearer and speaker, then only to hearer. However, for the sake of simplicity, and because the approximation can be done (we generally have the required conditions $N \gg 1$ and $t_{conv} \gg 1$), we will consider $u_0 = 0$ as initial condition. This gives us the solution:

$$u_t = N\left(1 - \left(\frac{N-1}{N}\right)^t\right) \tag{5.14}$$

Convergence for the meaning m can be defined as the first moment t_1 when the conventions has spread to all agents, i.e. when $u_{t_1} > N - 1$ [2]. This translates as:

$$t_1 = \frac{\ln(N)}{\ln(N) - \ln(N-1)} \approx N \cdot \ln(N) \tag{5.15}$$

[2] N itself cannot be reached, as it is an asymptotical value. Anyhow, for $u_t > N - 1$, considering u_t as continuous does not make sense anymore.

If this situation extends to all meanings – which is definitely a best case scenario – the global convergence time will be:

$$t_{conv} \approx MN \ln(N) \tag{5.16}$$

We can get the same result via reasoning similarly to the Coupon Collector's problem (Blom, Holst, and Sandell, 1994), formalizing the number of samples necessary to see at least once every item from a set sampled uniformly.

5.2.3 *Backpropagation of the information*

We have seen in the previous paragraph how information can spread from one agent to the whole population. The reverse question, but quite analog, is the following: how is information about the whole population gathered by an agent? After how many interactions does an agent know that its state has been influenced (potentially indirectly) by most agents in the population?

Let us consider an agent $\mathcal{A}$ and note u_t the number of agents that $\mathcal{A}$ has had information about after t interactions. After 1 interaction, obviously $\mathcal{A}$ has information only about its interlocutor and itself. $u_1 = 2$ After the second interaction however, it gathers information from the new interlocutor, which was influenced by its own previous interlocutor. In other words, $u_2 = u_1 + 2 = 4$; and $u_t = 2^t$.[3]

[3] One could say that interactions could happen with agents that are already in the set: however, before $u_t \approx N$, the probability of this happening stays low enough; and the relevant factor here is the order of magnitude to reach a significative sample of the population, not the whole population.

The average time needed, in number of interactions were $\mathcal{A}$ is involved, to gather relevant information about the population is :

$$t_{gather} = \lceil \log_2(N) \rceil \tag{5.17}$$

The sliding window of past interactions considered for the Coherence strategy in chapter 4.3.3 is using this value, adding 1 to potentially trigger success of the interaction.

5.2.4 *Statistical lower bound to the number of inventions*

Of course, it is highly improbable to be in the situation described above. The natural intuition is that at least one meaning should be involved in more than one invention. Let us first consider a situation with full-exploitation, i.e. exploration only when an agent's lexicon is empty. Because every agent has an equal chance to be speaker or hearer for its first interaction, there are in total $\frac{N}{2}$ inventions. The number of meanings corresponding to these inventions naturally satisfies $M_{inv} \leq \frac{N}{2}$. If we note u_i the number of meanings that have been involved in at least one of the first i inventions, we have:

$$u_{i+1} = u_i + \frac{M - u_i}{M} \tag{5.18}$$

And therefore, with $u_0 = 0$:

$$u_i = M \cdot \left(1 - \left(\frac{M-1}{M}\right)^i\right) \tag{5.19}$$

Which gives us:

$$M_{inv} = f(N, M) = u_{\frac{N}{2}} = M \cdot \left(1 - \left(\frac{M-1}{M}\right)^{\frac{N}{2}}\right) \tag{5.20}$$

We can conclude at this point that $\frac{N}{2} - M_{inv}$ inventions were done with meanings that had already been involved in a previous invention. After converging on all those meanings, agents will move to the remaining ones. With the assumption that this switching behavior happens at the right moment, we are faced with the exact same problem, only with a reduced number of meanings $M_1 = M - M_{inv}$. They will engage in the same process of inventing conventions for a part M_1^{inv} of the remaining meanings, and start again with $M_2 = M_1 - f(N, M_1)$. The progression in the space of meanings is done by *chunks* of meanings, of decreasing size, until all meanings have been involved. The size of the set of unexplored meanings obeys the following law, with initial condition $M_0 = M$:

$$M_{i+1} = M_i - f(N, M_i) = M_i \cdot \left(\left(\frac{M_i - 1}{M_i}\right)^{\frac{N}{2}}\right) \tag{5.21}$$

The number of chunks n_{ch} corresponds to the first index satisfying $M_{n_{ch}} < 1$. Having the number of chunks n_{ch}, we can derive the total number of inventions $N_{inv} = n_{ch} \cdot \frac{N}{2}$.

A few values are presented in table 5.1. These values, and more importantly their corresponding set $(M_i)_{0 \le i \le n_{ch}}$ were used in chapter 3.2.4, for the Chunks Decision Vector strategy.

N	M	$\left\lfloor \frac{2 \cdot M}{N} \right\rfloor + 1$	n_{ch}
20	100	11	11
200	1000	11	12
2000	10000	11	12
50	100	5	5
500	1000	5	5
5000	10000	5	6
100	100	3	3
1000	1000	3	3
10000	10000	3	3
200	100	2	2
2000	1000	2	2
20000	10000	2	2
500	100	1	1
5000	1000	1	1
50000	10000	1	1
1000	100	1	1
10000	1000	1	1

Table 5.1: A few values of the number n_{ch} of chunks of invented meanings. Values were computed by iterating equation 5.21 and with the stopping condition $M_i < 1$.

If $N \gg M$ **:** $n_{ch} = 1$ and $N_{inv} = \frac{N}{2}$

If $N \ll M$ **:** $n_{ch} \approx \frac{2 \cdot M}{N}$ and $N_{inv} \approx M$

If $N \approx M$ **:** $n_{ch} \approx \left\lfloor \frac{2 \cdot M}{N} \right\rfloor + 1$ and $N_{inv} \approx M + \frac{N}{2}$.

We will in all cases consider $N_{inv} \approx M + \frac{N}{2}$.

5.2.5 *Statistical lower bound to convergence time*

During a Naming Game, the dynamics of individual meanings m fall under two categories: they are either topic of a single invention of a convention, or several. In the first case, the minimum number of required interactions per meaning was calculated in paragraph 5.2.2, and its value is $t_1(N) \approx N \ln(N)$. In the second case, the dynamics are in a first phase similar to the Naming Game with $M = 1$, but with $N_{inv}(m)$ agents only; those who have invented a convention related to meaning m, converging in a time $t_{RTC}(2 \cdot N_{inv}(m))$. Both speaker and hearer are concerned in the same way at the time of invention, hence the factor 2. Interactions with other agents that might happen already during this process are counted as part of the second phase: spreading the winning convention to the rest of the population.

Again, the optimal dynamics of spreading can be described by the relation 5.13; the only parameter changing is the initial condition u_0.

If $N \approx M$ or $N \ll M$, there is on average only a single invention per meaning, which leads to a similar $u_0 = 1$. The maximum average number of inventions per meaning is obtained when $N \gg M$, and is equal to $\frac{N}{M}$. $u_0 = \frac{N}{M}$ gives us:

$$u_t = N \left(1 - \left(\frac{M-1}{M} \right) \cdot \left(\frac{N-1}{N} \right)^t \right) \tag{5.22}$$

With $M \gg 1$ and $N \gg 1$ the solution verifies $t_1 \approx N \cdot \left(\ln(N) - \frac{1}{M} \right) \approx N \cdot \ln(N)$.

We can eventually write the corresponding expression for a lower bound of the convergence time t_{conv}, as the optimal time t_{opt}, using the approximations of the previous paragraph and the relation $N_{inv}(m) \approx \frac{N_{inv}}{M}$:

$$t_{opt}(M, N) = M \cdot \left(N \cdot \ln(N) + t_{RTC} \left(2 + \frac{N}{M} \right) \right) \tag{5.23}$$

Of course, there can be more than $2 + \frac{N}{M}$ inventions for a given meaning m, the relation above is just a mean field approximation. However, if for m there are more than average inventions, its contribution will be more than average, because of the exponent $\frac{3}{2} > 1$ of the corresponding powerlaw. To illustrate this, let us take the example where all the N_{inv} inventions concern only one meaning, the second term of t_{opt} will then be equal to:

$$t_{RTC}(N_{inv}) \propto N_{inv}^{\frac{3}{2}} > M \cdot t_{RTC} \left(\frac{N_{inv}}{M} \right) \propto \frac{N_{inv}^{\frac{3}{2}}}{\sqrt{M}} \tag{5.24}$$

The mean field approximation leads to a lower bound of the value corresponding to the real scenario, and is therefore acceptable in our case.

5.3 *Performance measures*

How close to optimal behavior are our topic choice strategies? We will here define a few performance measures based on the lower bounds expressed in the previous parts of this chapter, to classify them and be able to compare them directly across different values of M and N. Each performance measure will be in the range $[0\ ;\ 1]$, 1 being the optimal value.

5.3.1 *Convergence time*

The ratio of convergence times (inverse to keep the value in $[0\ ;\ 1]$) reflects how close to optimal convergence time t_{opt} the observed convergence time t_{conv} is.

$$P_{CT} = \frac{t_{opt}(N, M)}{t_{conv}} = \frac{M \cdot \left(N \cdot \ln(N) + t_{RTC} \left(2 + \frac{N}{M} \right) \right)}{t_{conv}} \tag{5.25}$$

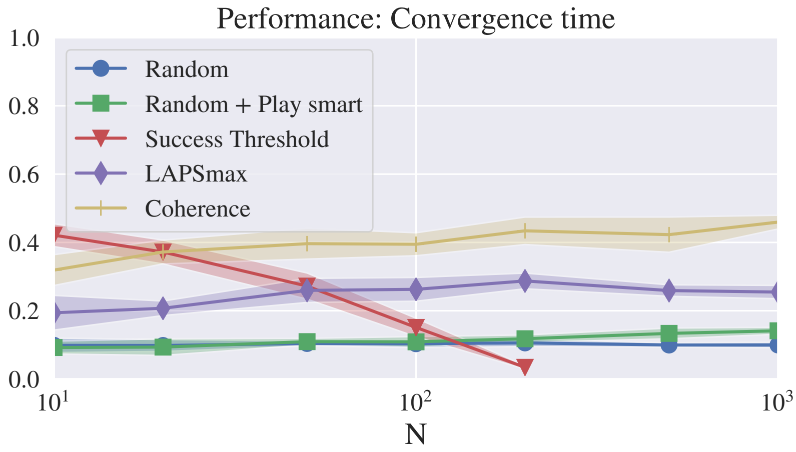


Figure 5.3: Performance measure for convergence time, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

We can see on figure 5.3 that Success Threshold has the best performance for the lowest values of N, but quickly drops as N approaches $2M$. On the contrary, Coherence and LAPSmax are stable or increasing with higher values of N, and Coherence even reaches 0.5. As t_{opt} is a lower bound, nothing ensures that values close to 1. can actually be reached. The *play smart* word choice policy starts to influence the Random Topic Choice only when $N \geq 2M$.

5.3.2 *Convergence speed*

We call speed the increase over time of the theoretical communicative success $S(t)$. In the optimal case, the maximum value 1 is reached in t_{opt} interactions. The performance value below reflects how close to optimal speed the observed dynamics are, by comparing the values of $S(t_{opt})$ in the optimal case and in the observed case.

$$P_{CS} = \frac{S\left(t_{opt}(N, M)\right)}{S_{opt}\left(t_{opt}(N, M)\right)} = S\left(t_{opt}(N, M)\right) \tag{5.26}$$

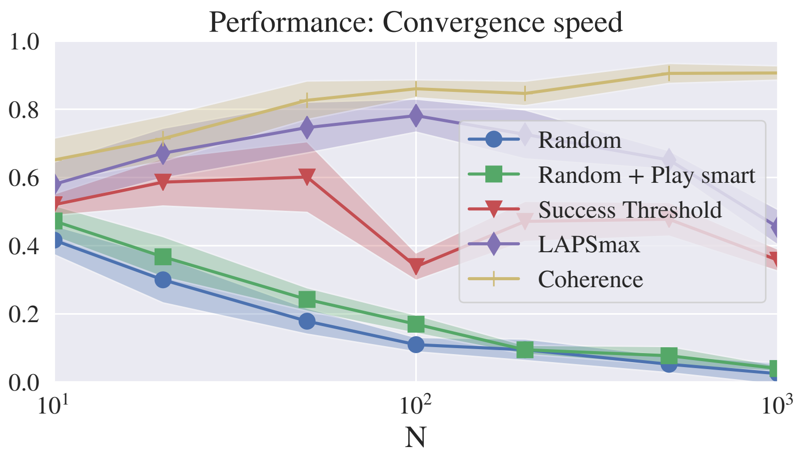


Figure 5.4: Performance measure for convergence speed, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

We can see on figure 5.4 that Success Threshold, although not converging within a reasonable time[4] for $N \geq 2M$, still quickly reaches high values for $S(t)$. This suggests that the system may have slower

[4] Less than 10^7 interactions.

dynamics in level 2.b of the ATC description found in 3.1.3. LAPSmax strategy has a high performance, but drops for high values of N. Coherence on the contrary dominates and seem to asymptotically reach 1. Both versions of RTC have similar behavior.

5.3.3 *Exploration*

Exploration can be characterized by the number of inventions. A natural performance measure is thus the ratio of inventions of optimal case and observed case (inverse to keep the value in $[0\ ;\ 1]$)

$$P_{EX} = \frac{N_{inv}^{opt}(N, M)}{N_{inv}} = \frac{M + \frac{N}{2}}{N_{inv}} \tag{5.27}$$

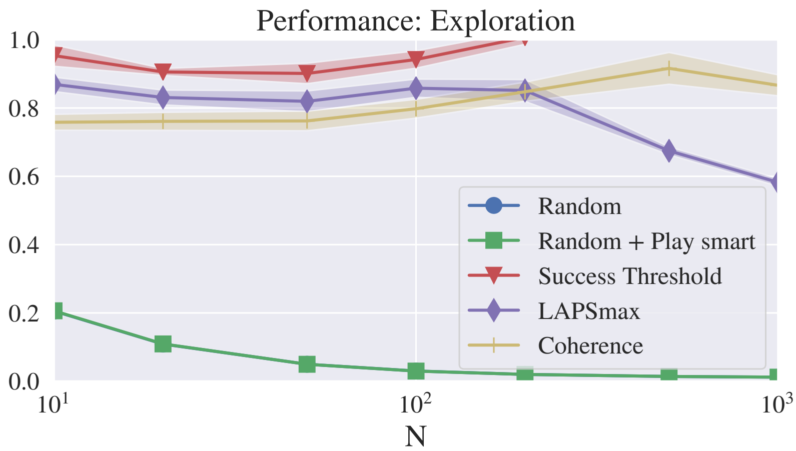


Figure 5.5: Performance measure for exploration, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

We can see on figure 5.5 that the behavior of all active strategies is similar and slowly decreasing for $N \leq 2M$, then LAPSmax is stable, and Coherence increases again. All perform significantly better than RTC strategies.

5.3.4 *Spreading*

We have distinguished exploration and convergence performances, but what if a stragey optimizes the spreading process, but not the exploration phase? By applying the same principles as for the optimal case, we can determine an optimal convergence time given a number N_{inv} of inventions:

$$t_{opt}^{inv}(M, N, N_{inv}) = M \cdot \left(N \cdot \ln(N) + t_{RTC}\left(\frac{N_{inv}}{M}\right)\right) \tag{5.28}$$

And derive from that the spreading time performance:

$$P_{ST} = \frac{t_{opt}^{inv}(N, M, N_{inv})}{t_{conv}} = \frac{M \cdot \left(N \cdot \ln(N) + t_{RTC}\left(\frac{N_{inv}}{M}\right)\right)}{t_{conv}} \tag{5.29}$$

We can see on figure 5.6 that the active strategies have a similar shape than for the convergence time performance measure: in

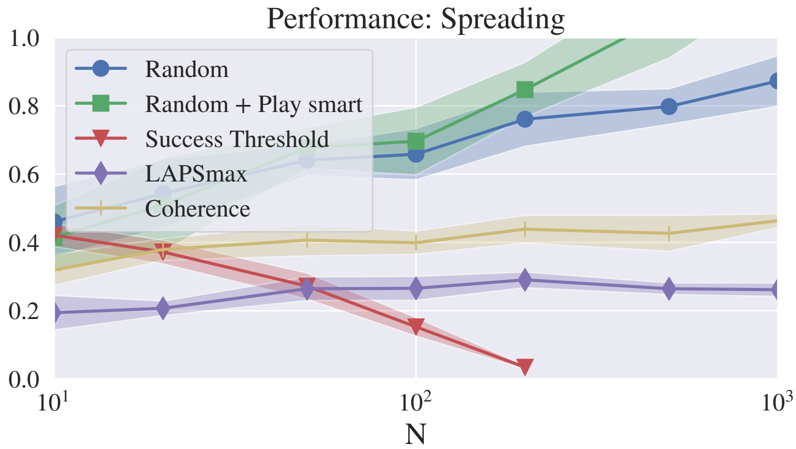


Figure 5.6: Performance measure for spreading, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

fact, with N_{inv} close to minimum, those two measures are almost the same. The two RTC strategies have high values, because their corresponding N_{inv} is high, and their convergence time $t_{RTC}(M, N)$ gets close to $M \cdot t_{RTC}(\frac{N_{inv}}{M})$. *Play smart* even goes above 1: The lower bound was calculated with the assumption of random word choice, the values for $t_{RTC}(N)$ used in the formula of the performance measure should consider this as well. This would constitute a direction for a future development.

5.3.5 *Spreading speed*

Again, as for P_{CS} the speed can be compared to the optimal case with the relation:

$$P_{SS} = S\left(t_{opt}^{inv}(N, M, N_{inv})\right) \tag{5.30}$$

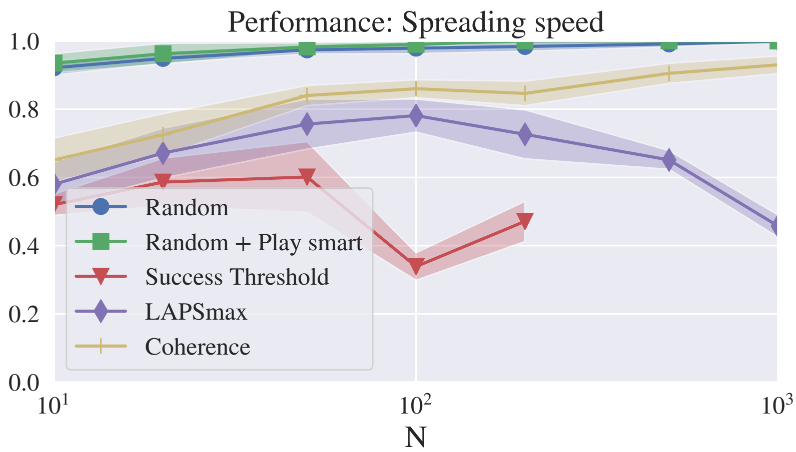


Figure 5.7: Performance measure for spreading speed, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

We can see on figure 5.7 results related to the previous measure: performance is similar to the convergence speed measure for active strategies, and RTC strategies are close to or reach optimal value 1.

5.3.6 *Lexicon size*

Last but not least, we can define a measure comparing the maximum average lexicon size to its minimal value M:

$$P_{LS} = \frac{M}{N_l^{max}} \quad (5.31)$$

Typically for efficient strategies, this value reaches 1. We can see on figure 5.8 that it is the case for active strategies, only LAPSmax starts to drop for $N \geq 2M$.

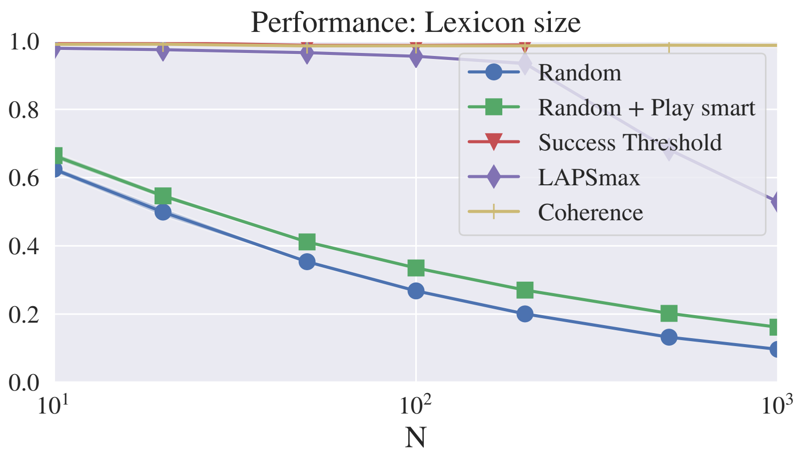


Figure 5.8: Performance measure for lexicon size, for several strategies. $M = 100$, $W = \infty$, averaged over 8 trials.

5.4 *Discussion*

We defined six criterias to classify ATC strategies, based on the estimation of a statistical lower bound for the convergence time.

The first performance measure, which is the one characterizing convergence time, relies directly on the statistical lower bound, and we do not know if this boundary is in fact reachable. However, we showed that at least the order of magnitude is right, as half of the value (performance 0.5) seems to be the asymptotical limit for one of the strategies (Coherence). Interestingly, this corresponds to a convergence time per meaning in $2 \cdot N \cdot \ln(N)$, which is the time to propagate one specific word to the whole population with Random Topic Choice[5].

We showed that Coherence is systematically best scoring for all the performance measures (if considered over all values of N), and has stable or increasing scores (with respect to N). It even stays at the maximum for P_{LS}, meaning that lowest possible memory usage (for the lexicon) is achieved. LAPSmax on the contrary starts to show a small decrease in performance around $N = 2M$. This is probably due to the choice $\tau = 2$, which should maybe start being higher for these values of N. This would allow to take into account more information from the past, which may be necessary as N increases.

We have discussed in the previous chapter the limit $N = 2M$ corresponding to a change in behavior, this can be seen directly on the

[5] A proof can be found in De Vylder, 2007.

performance measures for some of the strategies. This change in behavior also result in the apparition of the second term (equal to 0 before) in the expression of t_{conv}^{opt}.

This second term is still neglectable, within the parameter ranges that we chose[6]. It would be interesting to further the study to higher ratios of $\frac{N}{M}$ to study its impact. We could lower the value of M, but for low values of M (≈ 10), it may be difficult to distinguish statistical effects of low values with the impact of the second term. Simulations should be done for greater values of N (above 10^6), which would require more computational resources than what we are currently using, or to further optimize the code.

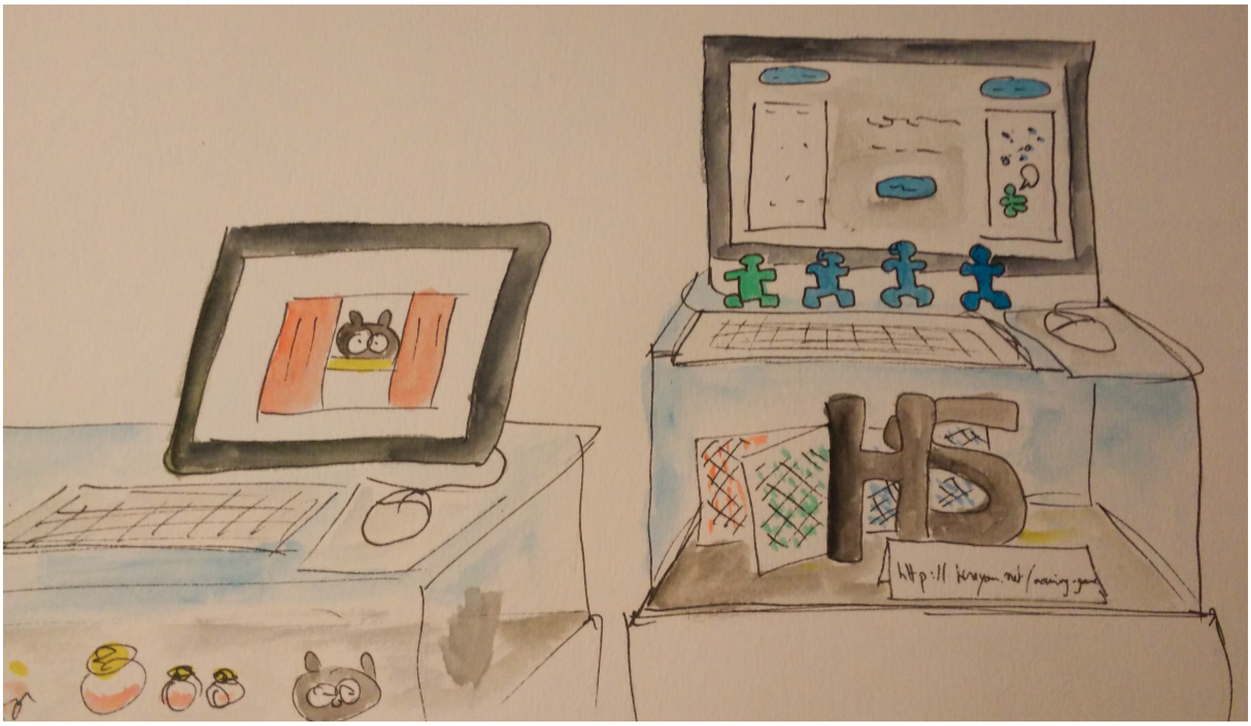
[6] For example for $N = 1000$, we should have $M \approx 5$ to have a symetric contribution from both terms. With $M = 100$, the contribution of the second term is $\approx 10\%$

We saw that the Random Topic Choice performs quite bad compared to other strategies, excepted for spreading and spreading speed: in fact, given a high number of inventions, the optimal behavior gets closer to RTC. In other words Random Topic Choice spreads conventions more efficiently, but because of the presence of too many conflicts fails to get good scores for the other measures. The difference between the two word choice policies for RTC is only slightly noticeable for most measures: this is because they only start to quantitatively differ when $N \geq 10^4$ (Baronchelli, Dall'Asta, et al., 2005). As for the spreading measure reaching values above 1 for *Play smart*, as explained in the main text this is due to the usage of convergence time associated to normal RTC. To retrieve 1 as upper boundary, one should consider a normalization by convergence times for *Play smart* associated to $M = 1$.

Computational Social Science Summer School, Sant Antioco, Sardinia. July 2017

Part III

Back to reality



Kreyon Conference, Rome. September 2017

6 Human Behavior

Contents

6.1 Introduction

Do real people control the complexity of their interactions, especially in the context of lexicon emergence?

A famous physicist joke[1] involves finding a solution to the diseased cows of a farmer. A physicist studies the problem and gives a solution *working for a billion spherical cows in vacuum*. The problem might be really interesting and the solutions found extremely elegant, they can still be far from being useful to the initial question.

Since the root of our problematic lies in reality, it is important to come back to it and assess the viability of the theoretical findings, assuring that the simplifying assumptions that were made and ranges of parameters that were studied are not *approximated cows*.

Experimental semiotics [2] is a way to do so, while studying the emergence and evolution of social consensus in real humans. We

[1] See Winckler, 2009 for a detailed review of scientific humor.

[2] See paragraph 1.3.2 for an introduction.

designed an experiment in this sense, inspired mainly by the work of Centola and Baronchelli, 2015 on the Naming Game. The idea is to have participants take the role of an agent within our model, and see whether they exhibit a pattern in their topic choice policies which would be different from random topic choice. From the start we considered recruiting participants online, either on crowdsourcing platforms or by making the experiment attractive, in the form of a game.

The experiment went through several phases of development: A first version was developed in 2017 for a presentation to the Kreyon Conference in Rome[3], where the public could try the experiment among several others. It was extended as a project of the Hack'1Cerveau hackathon in the Cap Sciences fablab of Bordeaux, to include the possibility of a multiplayer game[4]. The final version was brought up with the help of Sandy Manolios, as subject of her master's thesis.

[3] Conference on Innovation and Creativity, see kreyon.net

[4] The other versions being one real human interacting with simulated agents

In this chapter, we will detail the experiment that we designed, and show that first results clearly indicate that people tend to limit the number of inventions during the game. Software aspects[5] are to be found as appendix at the end of this manuscript A.

[5] Available as easily usable open source code on github: wschuell/ng_userxp.

6.2 Design of the experiment

6.2.1 Constraints

Our goal is to check whether humans do use mechanisms as active topic choice to control complexity growth during the negotiation of new conventions. We can do this by replacing agents from our model by real participants and record their behavior. We can then analyze the behavioral trace using relevant measures (detailed in paragraph 6.3.1) and compare the results to the theoretical trace of a passive behavior (i.e. random topic choice).

Compared to theoretical/simulated work, conducting a user experiment is subject to a certain number of new constraints:

1. The experience should not be frustrating for participants, on the contrary they should be motivated to complete the task.
2. It should not take too long for someone to take the experiment: strictly less than 20 minutes, ideally less than 5 minutes.
3. Participants should understand easily the context of the experiment and what they are asked to do.
4. The data produced by a reasonable number of participants should allow to draw conclusions.
5. The experiment should be conducted with a reasonable number of participants.

6.2.2 *Recruiting participants*

In many semiotic experiments (Centola and Baronchelli, 2015; Raviv, Meyer, and Lev-Ari, 2019; Verhoef, Kirby, and De Boer, 2014; Vollmer et al., 2014), participants are invited to physically come to the laboratory and perform a specific task. Conducting such an experiment needs a lot of time, and even space in the lab if participants are required to come by groups. Because of that, the number of participants stays low. Attempts to scale up the number of participants, e.g. by organizing a big event concentrating the participations, may as well fail as participants will not feel as committed to attend as in a small setting.

An alternative is to recruit participants online: they do not have to physically come to the laboratory,many participants can do the task at the same time and the experimenter's presence is not required. The experiment becomes easily scalable. Several crowdsourcing platforms are available to recruit a large number of participants:

Amazon Mechanical Turk: [6] This platform is the most famous, and allows to recruit people to perform online tasks. Its original goal is to deal with work that cannot be done by computers, but has to be scaled and performed by many people. The platform offers the possibility to ask for this type of work to be done as it would be to a computer: participants are automatically recruited and results collected. Participants are recruited world-wide, and are remunerated.

[6] The name comes from a tale where a man hidden in a machine was playing chess, giving the illusion that people were playing with a robot.

Foulefactory: A french equivalent to the Mechanical Turk.

Prolific.ac: This platform was built by and for academics, as an alternative to the Mechanical Turk. Remuneration is controlled to be ethical considering european standards.

CrowdCurio: A platform collecting online research experiments. Participants are volunteers, and their recruitment is more based on interest and willingness to contribute to science. Focus on the domain of curiosity.

XTribe: Similar to CrowCurio. Focus on the domain of creativity and innovation.

Platforms usually rely on the experimenter putting up your experiment on a website, and provide participants with the link. Participants can be screened to fit certain criterias. The platformhas to be notified when a participant finishes their task. Of course, in this configuration, promotion and recruitment of volunteers through social media can be done in parallel.

To recruit participants without remuneration, or to keep them motivated, experiments can be presented as games (Morin, Winters, T. F. Müller, et al., 2018), which also has the advantage of potentially reaching a great number of participants without needing proportional funding. This is for example the strategy chosen by the Color Game (Morin, Winters, T. Müller, et al., 2018), where participants are players of an online app involving a communication task.

In our case, we designed the experiment in the form of a game, each run of the game being an instance of the Naming Game. We collected data first at a conference open to the public (Kreyon Conference in Rome, 2017), and later through advertising on social media with a second version of the interface. Each dataset has around 80 individual games. Participants may have done more than one game, but usually not more than three. To scale up and check consistency on a larger sample of participants, a third set of data will be crowdsourced through *prolific.ac*.

6.2.3 *Interactions*

Compared to Iterated Learning, communication games like the Naming Game involve several interacting agents at the same time. Iterated Learning tasks can be crowdsourced easily, as individual tasks do not require to wait for another participant's action. Waiting for the others to finish their respective task can result in a high dropout of participants. If some suggested that people could be provided with entertaining unrelated content while waiting (e.g. cat videos), this solution would still not be robust for the number of participants required by the Naming Game (at least 4 or 5). Solutions that do not require participants to stay connected include asking them to reconnect within 24 hours (Schaekermann et al., 2018); or separating the production and understanding parts of the communication game and making them available to all players[7] (Morin, Winters, T. Müller, et al., 2018).

[7] In practice, a connected player can either produce a set of signals to be interpreted later, or pick the sets of signals that have been produced earlier by other participants.

In our case, the first solution is not feasible: one experiment should include at least a dozen data points per participant, which would result in a two weeks-long experiment, provided all participants do reconnect everyday, and in an adapted order. The second solution has another drawback: participants are not picked randomly but proportionally to their presence online. We relied instead on another solution: simulating the other agents. Each participant has the impression of interacting with other people, but without latency. Of course, the global properties of the game depend on the behavior of all agents, and the overall dynamics will only slightly be modified by the participant's behavior; but as our work focuses on the local behavior of the participants, this does not matter.

Interactions should be short (a few seconds) and not too numerous (around 20 maximum).

6.2.4 *Structure of the experiment*

The experiment is organized in a succession of screens, presenting different information to the participant, and letting them interact via buttons. For a detailed view of the organigram of the screens and their transitions, see figure 6.1. The different screens are:

Home: When connecting to the website, the participant lands on this screen. It presents basic information: language (possibility to change), identifying charachter for the participant, a logout button, and a button to start the game, which leads to the *Information* screen.

Information: More detailed information about the context: backstory, notion of interaction, roles (speaker, hearer, or waiting), notion of feedback (success or failure). A button to start the game, leading to one of the three interaction screens: *Speaker*, *Hearer* or *Waiting*.

Speaker: The participant has the role of the speaker for this interaction. This screen informs the participant of their role, and offers a first choice between the M meanings, presented as pictures. After this choice, the W words are displayed and the participant can pick one, and eventually send this information by clicking on a button, leading to the *Feedback* screen.

Hearer: The participant has the role of the hearer for this interaction. This screen informs the participant of their role, and of the word that was chosen by the speaker. The M meanings are displayed, along with an *I don't know* option: after picking one the participant can send the information by clicking on a button, leading to the *Feedback* screen.

Waiting: The participant is not involved in one or several interactions (if more than one consecutively, this nulmber is indicated). A button is available to proceed to the next interaction, and will lead to either *Speaker* or *Hearer*. If more than $T_{max} = 50$ interactions have been done, this same button leads to *Result*.

Feedback: After *Speaker* or *Hearer* screens, the participant is informed of the outcome of the interaction: failure or success of communication. In both cases the meaning initially selected by the speaker and the meaning understood by the hearer (potentially *I don't know*) are presented. A button is available to proceed to the next interaction: *Speaker*, *Hearer* or *Waiting*. If more than 50 interactions have been done, this same button leads to *Result*.

Result: This screen ends the current game and shows a final feedback about the level of agreement of the whole population, in the form of a score proportional to $S(T_{max})$. A button leads back to *Home*.

The exact instructions given to the participant can be seen in figure 6.4 for the first version, and in the appendix for the second version.

6.2.5 *Setting parameters*

To include the participant in a Naming Game, we need to define its parameters: population size N, number of words W, number of meanings M, maximum number of interactions T_{max}, and the behavior of the other agents.

Population size should not be too high, or the number of interactions to reach at least one success would be too high. It also should

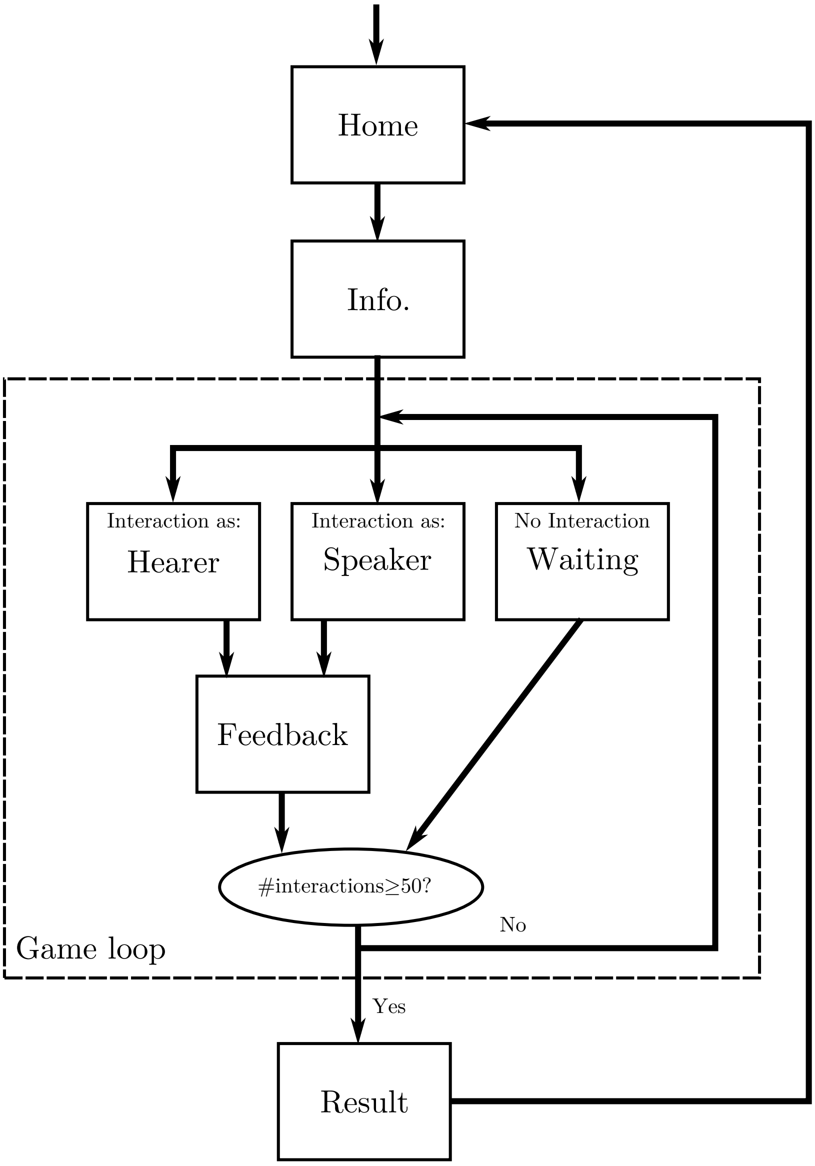


Figure 6.1: Structure of the user experiment: the different screens are represented by black rectangles, and the transitions between them by arrows. The participant arrives on a *Home* screen, showing some information and a button to start playing, leading to an *Information* screen, explaning a few details concerning the context of the game. The rest of the game is a loop: for some interactions the user is not involved, indicated on a screen *Waiting*; for the others they are either *Speaker* or *Hearer*. As a speaker, the participant chooses a meaning and a word. As a hearer, a word is provided, and the participant picks a corresponding meaning. After the respective choices, both screens lead to a *Feedback* screen telling them the result of the interaction: Failure or Success. If a count of $T_{max} = 50$ interactions has been reached, the game stops and a *Results* screen shows the final score (level of agreement, total and per meaning). If the count has not been reached, a new interaction starts and the next screen is again one of the three possible roles: *Speaker*, *Hearer* or *Waiting*.

not be too low, otherwise global agreement would be reached in only a few interactions and having a strategy would not matter. We chose the value $N = 5$.

The number of meanings should not be too high, otherwise participants may not differentiate them easily, let alone remembering the information associated to each meaning. It should also not be too low, as we focus on exploration of new meanings. We chose the value $M = 5$.

The number of words should be at least equal to M. A value a bit higher than M adds a degree of freedom: with $W = M$, homonymy would be necessarily associated to synonymy. A high value would have the same drawbacks as for M: confusion between words, and not remembering well associations with meanings. We chose the value $W = 6$.

The number of played interactions should not exceed 20, but a reasonable level of agreement should still be reachable (above 60% on average). The number of interactions per player (as either a hearer or speaker) is $\frac{2T_{max}}{N}$[8]. With the population size $N = 5$, this gives us $T_{max} = 50$. The dynamics of such a configuration can be seen on figure 6.2.5.

[8] Each interaction involves two agents: speaker and hearer.

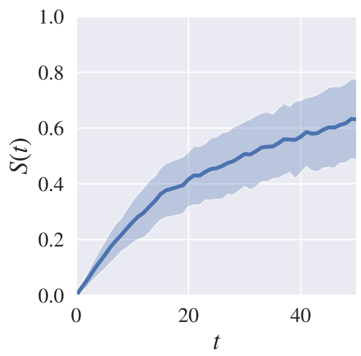


Figure 6.2: Communicative Success for Random Topic Choice with the parameters chosen for the experiment: only regularizing synonyms, $M = 5$, $W = 6$, $N = 5$, $T_{max} = 50$, averaged over 100 trials.

Other agents (simulated) follow a Random Topic Choice policy, and only regularize synonyms, not homonyms. This last modification was used to slow down the process and avoid reaching consensus too quickly.

6.2.6 *First version: Kreyon Conference*

The first version of the game was developped for the Kreyon Conference in Rome, which was open to the public on the two last days. This conference hosted a few other experiments, freely accessible to the public on computers set up for the occasion (see figure 6.2.6). Elisabetta Falivene helped to design the HTML skeleton of the application and Théo Segonds helped to deploy the application online. They both provided really useful advice and knowledge.

Figure 6.3: Experiment at the kreyon Conference in Rome: computers running several types of experiments were freely accessible to the public.

Meanings were represented as picture of common objects (a flower, a hat, water, fruits and bread), and words were randomly generated as a sequence of three syllables, each being composed of one consonant and one vowel. It was not possible for participants to write their own words, as they maybe would stick to it, or think that other agents in the game would also be people and understand english words. The information that has to be presented and understood by the participant include:

1. There are several agents interacting
2. They all start, as the participant, with empty lexicons
3. There is no preexisting language
4. A learning phase is necessary before reaching success
5. The choices of the participant can influence the global dynamics

6. The participant can have varying roles depending on the interaction (speaker, hearer, not involved)
7. Information collected at past interactions is relevant for next choices.

An illustrated example of an interaction was shown to participants (see figure 6.4 and appendix for a complete set of all screens), and a tutorial (or basic) version of the game was suggested to participants, in order to learn by playing. This version has lower values for a few parameters: $N = 3$, $M = 2$ and $T_{max} = 10$ to allow fast convergence and quick positive feedback.

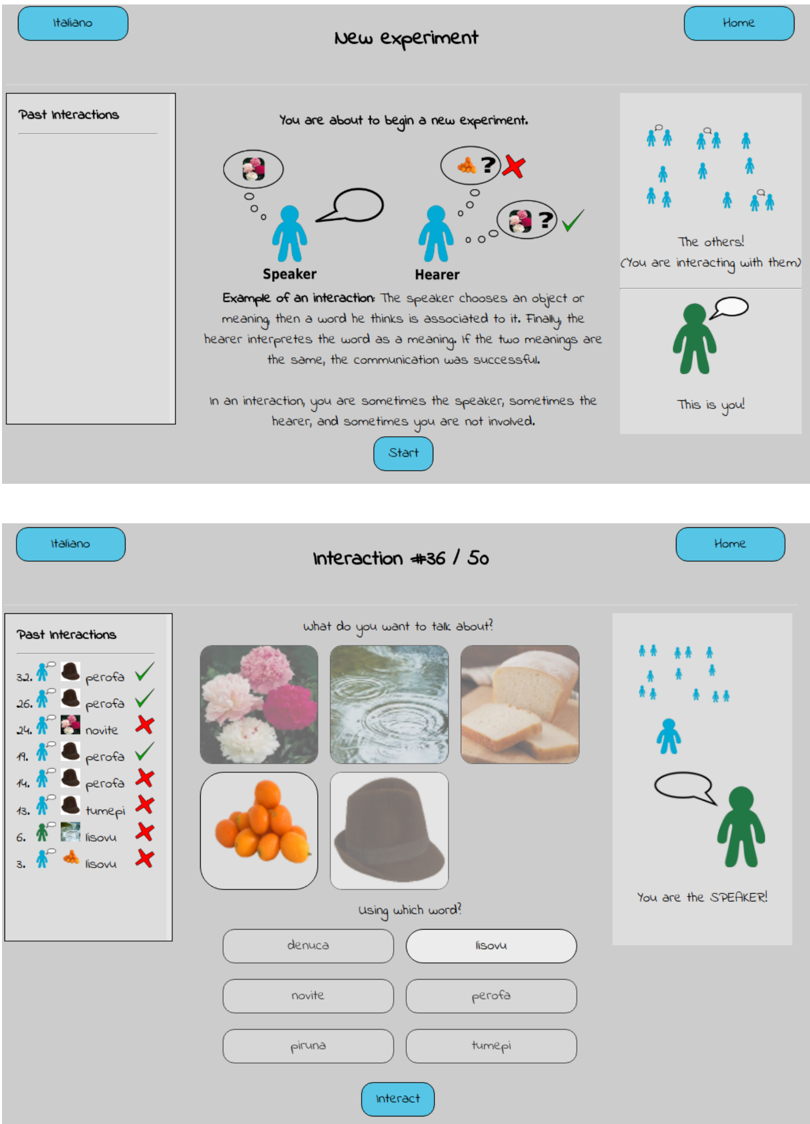


Figure 6.4: Interface of the first version. Upper image: Information screen. Lower image: Speaker screen. See appendix for a complete set of all screens from both versions of the experiment.

A certain number of elements were implemented to add attractivity and clarity to the experiment:

Identification of the participant's character: The participant was represented as a green little character, all other agents being blue.

Login: Possibility to be identified as a pseudonym and reconnect later. Along with the previous element, it is part of a gamification process, where attractivity of the game is increased by indentifying the player in the game.

History of past interactions: To avoid having to memorize information from the past interactions, they are summed up in a side column. Each past interaction is associated with the role of the participant (speaker or hearer), the meaning and word used by the speaker, and the feedback (failure or success).

Illustration of the interaction role: A right-side column shows a picture with the whole population, and the participant interacting or not, as a speaker or hearer.

Best scores: On the Home screen, a right side column shows the best scores so far (only for players who connected from the same machine).

Speaker items: The words only appear after the participant (as speaker) chooses a meaning. This avoid presenting too much new information on the screen, and focuses the attention of the participant on the set of meanings.

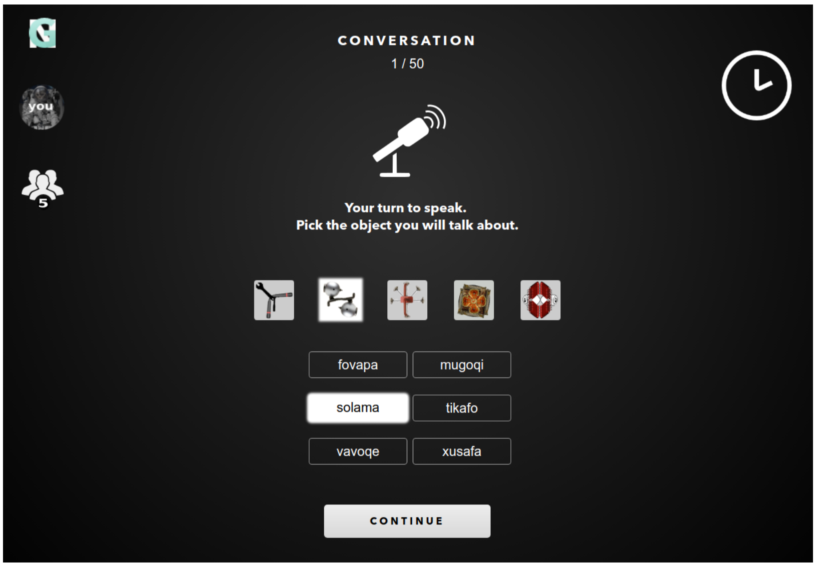
This version yielded a lot of feedback from the users: many dropped after a couple of interactions because they did not identify the context of the game, and many were frustrated by the strongly negative feedback provided by the red cross (presented when communication failed). Seeing this, to make them stay focused on the game and try to finish it, many people were told what underlies the dynamics, sometimes even a few interactions were played for them as a demonstration. The data collected in this way is therefore biased, and cannot be considered as the only source for an analysis.

We collected 71 distinct completed games. During the processing of the data, participants were not identified. Each participant may have completed several games. Not all participants were recruited during the conference: as the link was accessible for almost a year, some connected after, as well as some labmates and friends who wanted to try it as well.

6.2.7 *Second version*

Given the remarks of the last paragraph of the previous section, the experiment had to be redesigned, to avoid frustration, trigger more motivation, and be more understandable. For this part, we received the help of Sandy Manolios who did her master's internship on the topic. She helped to spot the elements that were missing for a more efficient gamification, and implement them in the new version. Atlal Boudir helped for graphical design and simplification of the interface. Théo Segonds helped again for the deployment and load-testing of the experiment on the servers.

Improvements in the second version include:



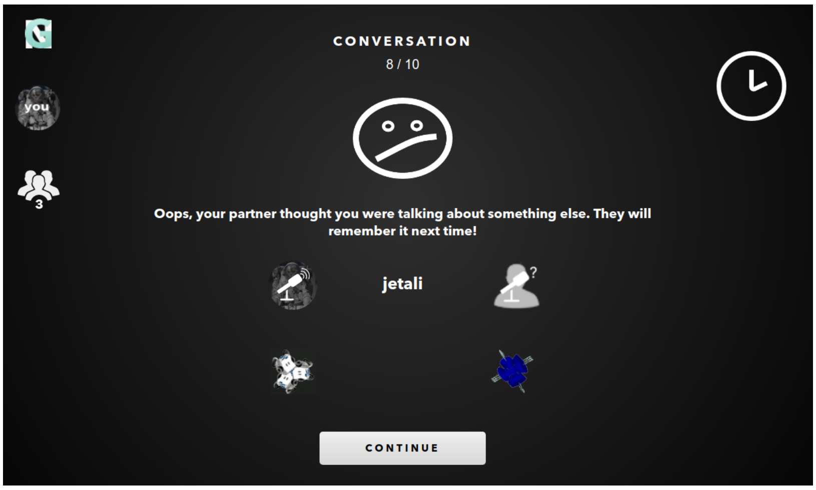


Figure 6.5: Interface of the second version. Upper image: Speaker interaction screen. Lower image: Feedback screen. See appendix for a complete set of all screens from both versions of the experiment.

Backstory: Providing context helps to understand the game and the goal of the interactions. As in the experiments of Verhoef, Kirby, and De Boer, 2014, the participant is told that they are interacting with aliens. In our case, aliens composing the group are from different planets, and need to build up a common language from scratch to escape the spaceship they are trapped in. The few paragraphs presenting the story also link the different concepts with pictures: the player's character, the other participants and their number (as a numbered icon), and the icons associated to the different roles (speaker, hearer, not involved).

History of past interactions: The column of the first version represented too much visual information. The new interface provides this information only when clicking on a clock button. Also, only information acquired as hearer is provided, it corresponds to passive assimilation and allows us to remove one item from each past interactions: the role, as it is in this case always hearer.

No important information on the right side: A right-side column is almost always ignored by intuitive visual exploration.

Success and failure pictures: Not anymore a red cross and a green check, but a happy face and a perplex face. There is still feedback, but less frustrating.

Meanings: The picture of the first version were known objects, and people did not understand why they had to come up with new words when they already had a word for those object (in their mothertongue). Unknown objects, but still recognisable, were chosen instead: those used in K. Smith, A. D. Smith, and Richard A Blythe, 2011 [9].

[9] The same pictures were reused in Verhoef, Kirby, and De Boer, 2014. We gratefully thank Kenny Smith for providing those images.

Locked levels: At the beginning, only the tutorial is accessible. After finishing it, the participant can access the main game. A third level is unlocked only after finishing the game (condition of success: reach an agreement of 65%, happening half of the time). This third level is only a bonus page, with few information about the design of the experiment and the scientific background, as well as a quick survey[10] Having something unknown to unlock can be efficient to keep participants motivated by the task.

[10] Data from the survey was not analyzed: too few people actually filled it. The crowdsourced version of the game on *prolific.ac* will solve this, as it will be a mandatory requirement to finish the task.

Interactive bubbles: containing relevant but short information, appearing when hovering with the mouse over an item (population icon, character icon, etc.).

Feedback was positive compared to the first version, and the game spread quickly through social networks, even if affluence dropped after a few days. A couple of messages (which were shared a certain number of times) triggered almost all the data available.

We collected 85 distinct completed games, done by 47 individual participants.

6.3 Results

6.3.1 Measures

Our aim is to determine whether the average behavior of participants corresponds to an active topic choice, or is closer to random. To be able to do this, we need a few measures, listed below. The input for those measures is a sequence of interactions done by the participant, including role (speaker or hearer), feedback (failure or success), and the meaning and word used by the speaker. Following the stream of interactions, we can determine whether a chosen meaning m when the participant is a speaker was already in the participant's lexicon or not. If it was not, the participant has explored during this interaction, and it is marked as an *exploration event*.

Scores: The obtained score (between 0 and 100), which is the value of the communicative success $S(t)$ at the end of the experiment.

Number of inventions: Number of inventions per experiment, done by the participant. As we have seen in previous chapters, limiting inventions is the key feature of active topic choice policies.

Exploration rate: The ratio of exploration events among all interactions where the participant is a speaker. This value is directly linked to the number of inventions[11], and is kept low in active strategies. Considered per lexicon size [12], the $M+1$ corresponding values[13] is to be compared with the *Decision Vectors* introduced in paragraph 3.2.1. As the vectors are well-defined for baseline strategies (Random Topic Choice and Exploration Biased Topic Choice), we will study exploration rates under this form.

Success threshold: We can compute the ratio of successes and failures in the same way as for the Success Threshold strategy, for every exploration events.

Min. Counts: Same as above, but for the absolute count of successes, used in the Minimal Counts strategy.

LAPS: LAPS value during exploration event. We normalize this value so it can be compared between different lexicon sizes (normalization factor $\frac{|\mathfrak{M}|}{|\mathfrak{M}_k|}$) and it has always a maximum of 1.

Coherence: Coherence value when exploring, as for the Coherence strategy.

[11] For a given experiment, the exploration rate is the ratio of the number of inventions over the number of interactions involving the participant as a speaker in this experiment.

[12] i.e. counting exploration events among all interactions where the participant both is a speaker and knows a given number of meanings.

[13] For lexicon size from 0 to M.

6.3.2 Description of the analysis

The measures themselves do not serve any purpose: we need to compare them to their values when a known strategy (possibly Random Topic Choice) is used. To do this, we built a small database of experiments where the participant is also simulated, having a given strategy. Studied strategies include: Random Topic Choice, Exploration Biased, Success Threshold (with a threshold of 50% given the small number of meanings and agents), Minimal Counts (with a threshold of 1), LAPSmax (time scale $\tau = 2$) and Coherence (time scale $\tau = 3$).

We did 80 simulations for each configuration, to be comparable with the available data from both versions of the experiment.

6.3.3 Scores

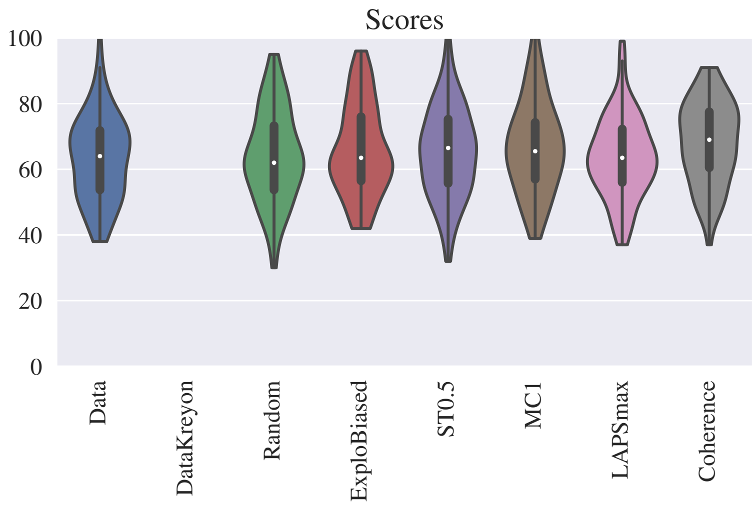


Figure 6.6: Comparison of scores for user data and several simulated strategies. Data for the first version of the experiment (DataKreyon) was not recorded. Data: 85 experiments, DataKreyon: 72 experiments, simulations: 80 experiments for each configuration.

If we look at the distribution of scores on figure 6.6, there is no significative difference between the different strategies. No conclusion can be drawn from this measure. Data for the first version of the experiment (DataKreyon) was not recorded.

6.3.4 Number of inventions and exploration rate

The number of inventions on the contrary shows a pattern that distinguishes the different strategies (see figure 6.7): we can see the clear tendency towards less inventions of active strategies. The exploration rate per lexicon size corroborates this hypothesis (see figure6.8). Data from both versions of the game show similar results, and have smaller values than Random Topic Choice, but still high compared to most active strategies. They seem to agree well with Exploration Biased strategy.

The previous result is corroborated by the exploration rates for each lexicon size, that can be seen on figure 6.8. We can also see on this figure that Coherence strategy explores only if the lexicon is empty; which is inevitable.

6.3.5 Strategy parameters

If we compare the values of the criterias for deterministic strategies (ST, MC, LAPSmax and Coherence, seen on figure 6.9), the pattern exhibited by the data of both experiments is again different from Random Topic Choice, and close to the distributuions associated to Exploration Biased. One exception can be noted: the LAPS measure, where user data is closer to Random Topic Choice. For each

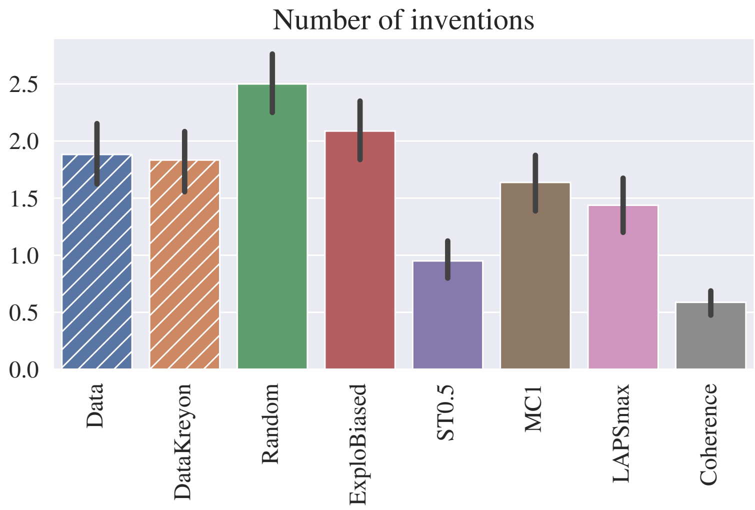


Figure 6.7: Number of inventions for user data and several simulated strategies. Active strategies invent less than Random Topic Choice, which is also the case of user data for both experiments. Data: 85 experiments, DataKreyon: 72 experiments, simulations: 80 experiments for each configuration.

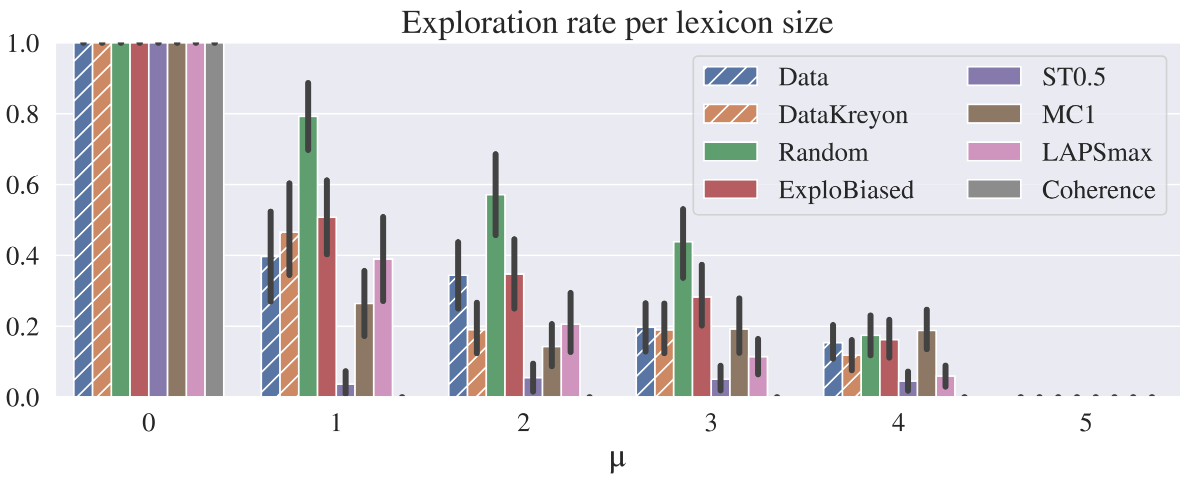


Figure 6.8: Exploration per number of known meanings μ, data and simulated strategies. Active strategies explore less than Random Topic Choice, which is also the case of user data for both experiments. This can be related with *Decision Vectors*, described in paragraph 3.2.1. Data: 85 experiments, DataKreyon: 72 experiments, simulations: 80 experiments for each configuration.

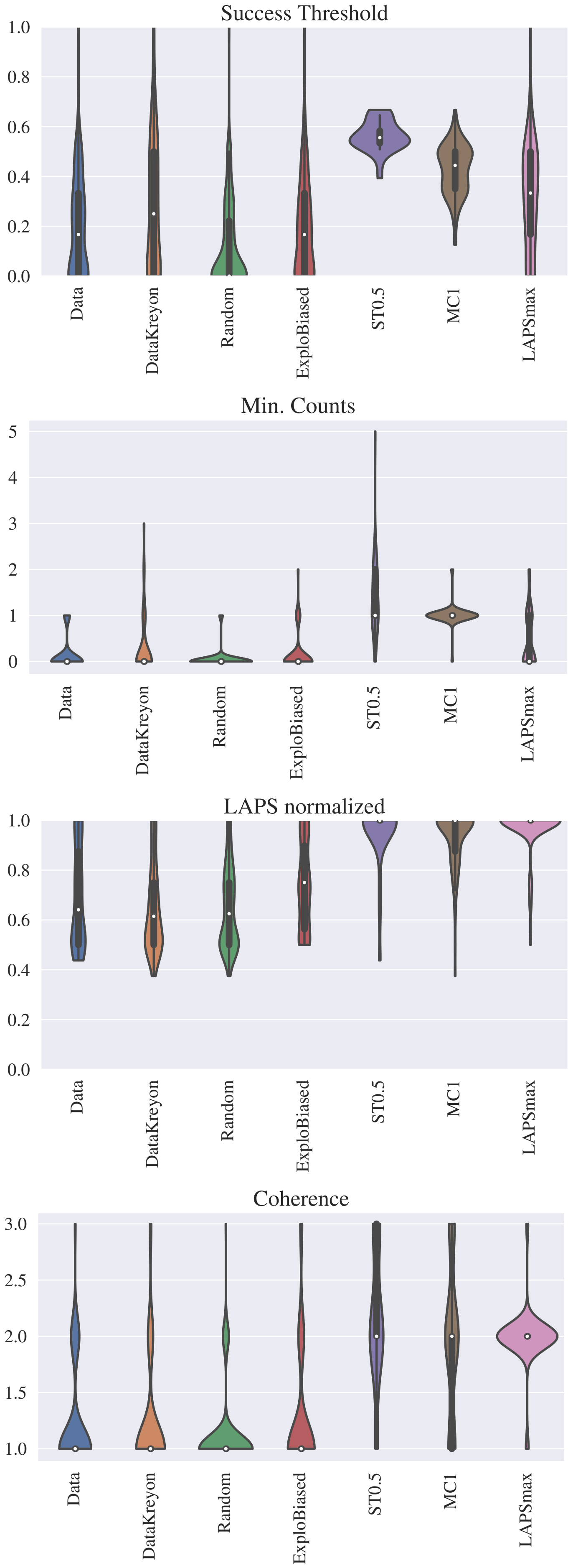


Figure 6.9: Parameter values from different strategies associated to exploration events for user data and several simulated strategies: Success threshold, minimal counts, normalized LAPS measure, and Coherence measure. See paragraph 6.3.1 for a description of the measures. The Coherence strategy is not represented, as no data is available (no observed exploration event). For the strategies depending on one of these parameters (ST, MC, LAPSmax and Coherence), we can retrieve the value by observing the corresponding figure. Coherence is not represented, as no exploration event has been observed in the simulated data. Data: 85 experiments, DataKreyon: 72 experiments, simulations: 80 experiments for each configuration.

deterministic strategy depending on one of these parameters, we can clearly retrieve the corresponding value (0.5 for Success Threshold, 1 for Minimal Counts, and 2 for LAPSmax.).

6.4 Discussion

We designed a user experiment placing participants in the context of the Naming Game in the form of a web application. We recorded their decisions, in particular when they decide to invent new conventions. We simulated a population of agents for each participant, so as not to depend on other participants finishing their respective task. A first version was presenting only the Naming Game and its rules, a second version was gamified and introducing a background story to get people motivated to finish the game, as well as understand it better. We defined a few measures to compare the obtained behavioral traces to several of the strategies introduced in the second part of this thesis: Random Topic Choice, Exploration Biased, Success Threshold, Minimal Counts, LAPSmax and Coherence. Measures include the number of inventions, the probability of exploring depending on lexicon size, and the values of the different confidence parameters associated to each deterministic strategy.

We showed that there is a clear tendency to control inventions (compared to Random Topic Choice). In terms of inventions and exploration, the average user behavior seems similar to Exploration Biased strategy, a bit more exploratory than other deterministic strategies. Exploration Biased is a really simple policy, and is still efficient in terms of convergence speed.

The average values of confidence parameters were measured as well, and for simulated strategies depending on one of those parameters, each corresponding value could be easily retrieved.

The next step will be to crowdsource the experiment on *prolific.ac*, and collect more data to confirm these results. Information collected in an end survey (with a mandatory completion for participants) could help to build more evidence.

However, if we want to characterize strategies and identify some of them [14], we do not need more data in the form of more experiments, since the strategy for each experiment may be different; but more data in the form of more interactions per game and maybe more meanings. A modification to the game that could make it last longer (in number of interactions) is to stop only when consensus is reached, and change the score to a *minimum convergence time*. We can as well reintroduce overall best scores, for example being displayed in the *Bonus* screen.

[14] Using criterions such as the Bayesian Information Criterion or BIC, typically used to compare likelihood of different models given a set of data points.

Increasing the number of meanings M could be more problematic, as a natural limitation of working memory could appear. However, this could also be studied through the same framework. Another possibility is to slowly introduce new meanings in the system, as an *expanding space* of available meanings.

Both features can be introduced in a multiplayer version, were participants actually have an impact on the experience of others. They cannot directly interact, as was argumented in paragraph 6.2.2. An idea, introduced as a project for a hackathon and already partially implemented, is to regularly exchange simulated agents between participants. Each participant has its own population of simulated agents to interact with, but regularly either new agents are created, or agents are exchanged with a global pool. Agents from other participants that are taken from the global pool remember what was learned with the first player, and participants influence each other in this way, without having to all be connected at the same time. This could also be used for other interactive Art & Science exhibitions, letting people build a language not just on their own but also with fellow visitors over a few days. This idea was already exploited before with the *Talking Heads Experiment* (Steels, 2015) where people were interacting with robotic agents and influencing their language over the course of several months, and later with an artistic purpose as part of an interactive exhibition at the *Fondation Cartier pour l'art contemporain* in Paris (figure 6.4), also with robots.

Figure 6.10: A language game as part of an interactive exhibition *Mathématiques, un dépaysement soudain* at *Fondation Cartier pour l'art contemporain* in Paris, 2011. This part of the exhibition was the result of a collaboration between Pierre-Yves Oudeyer, David Lynch, and Mikhaïl Gromov.

Visiting Ramon Ferrer-i-Cancho, UPC, Barcelona. August 2015 and November 2017

Part IV

Conclusions and Perspectives

Visits to Sony CSL, Paris. December 2017-February 2018

7 Conclusion

The main goal of this thesis was to study the impact of intrinsic motivation in the Naming Game, in the particular domain of active topic choice policies. These policies allow agents to actively control the growth of their lexicon. After reviewing many variants of the Naming Game model, we presented a new classification for them. We pointed out the issues that can be encountered with the model, in particular the burst of memory happening before reaching consensus.

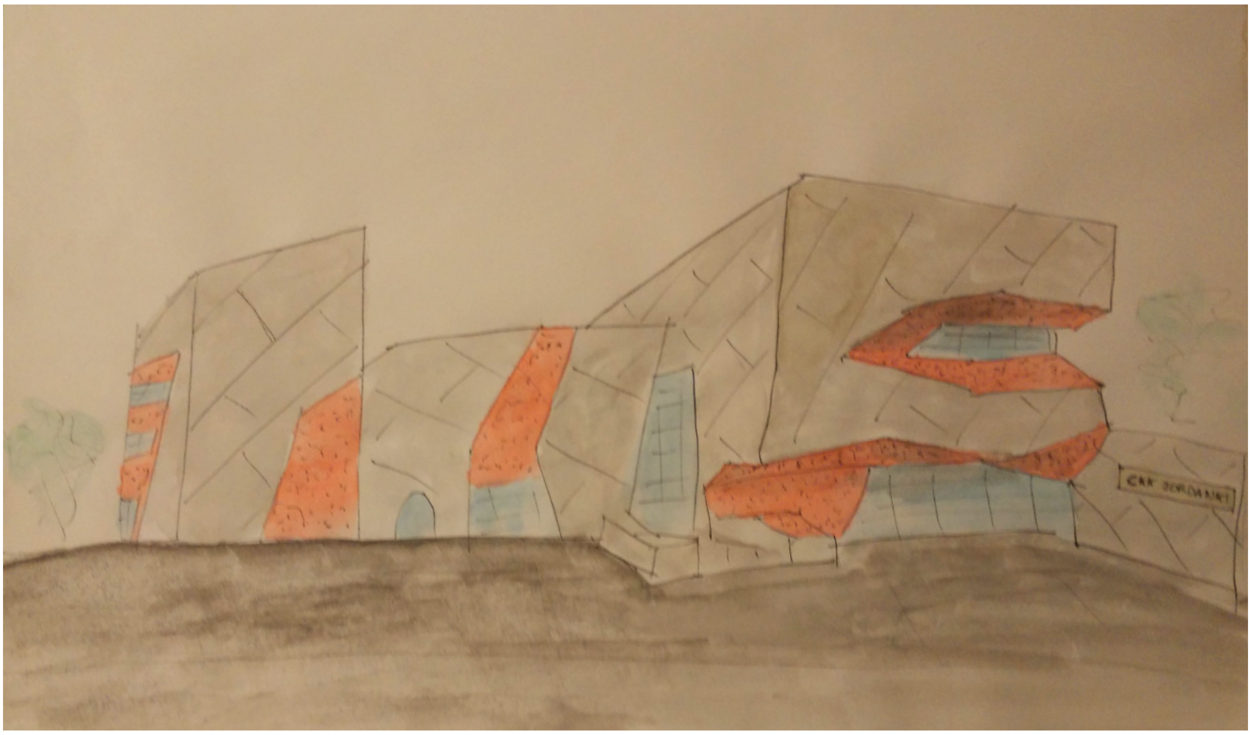
We identified topic choice policies as a way to introduce intrinsically motivated active choices in the Naming Game, and designed several strategies relying on different mechanisms and memory needs. Under hard memory constraints [1], Chunks, Success Threshold and Minimal Counts strategies show convergence properties that are close to the original model with random topic choice. Without artificially constraining memory, we defined the LAPSmax and Coherence strategies, both based on a representation of the average population vocabulary. The latter strategies do not only prevent the memory burst, but converge significantly faster than the original model. Moreover, they are robust to constraints on the space of words (introducing the possibility of high degree of homonymy).

[1] without synonymy and homonymy, thus preventing artificially the memory burst.

A theoretical approach allowed us to define a statistical lower bound to convergence time and several measures of performance to compare topic choice policies. Using those measures, we showed that the Coherence strategy is not only stable, but close to optimal over a wide range of population sizes.

We finally designed a user experiment, to place people in the situation of the model and record their decision concerning topic choice. We showed that there is a clear tendency towards limiting invention of new conventions, which is the key mechanism found in all the strategies that we defined before.

We showed that actively controlling the rate of invention of new conventions can result in faster agreement with less memory using computer simulations, and that people use this kind of mechanism as a natural behavior.



Evolang Conference, Toruń, Poland. April 2018

8
Future perspectives

8.1 Hearer's Choice

Hearer's Choice was introduced in paragraph 3.4.3 as an alternative scenario for topic choice: the hearer picks the topic, and not the speaker.

We have seen in chapters 3 and 4 that this scenario has some interesting properties: it reaches convergence in a time comparable to the standard topic choice scenario, but fails to control the number of inventions as efficiently, resulting in a moderate complexity burst. However, an interesting aspect is the dynamics at the end of the agreement process: there is no slow down like in the standard scenario. In other words, Speaker's Choice is more efficient to bootstrap the system and control the first wave of inventions, but for a later phase of *fixation* of the conventions, Hearer's Choice becomes more efficient.

To take advantage of both aspects, we could imagine a third scenario, where agents negotiate first who is going to choose the topic. At population level, this decision should be based on the average agreement level, i.e. the theoretical communicative success: low values would favor speaker choosing, and high values hearer. This can in fact be implemented at the agent level, using either the LAPS measure and an appropriate threshold, or criterias like lexicon size.

The resulting dynamics could have higher performance values than known strategies: At theoretical minimum convergence time, Coherence strategy reaches a communicative success really close to 1, but later the same amount of interactions is needed to reach complete convergence. As Hearer's Choice improves this part of the dynamics, the mixed scenario could be a way to get closer to the theoretical lower bound for convergence time.

It can be noted that the negotiation happening in the mixed scenario is another type of active behavior, that can be intrinsically motivated.

This alternative scenario can be reimplemented easily in the existing framework, aside the existing Speaker's Choice and Hearer's Choice scenarios, and reuse everything already in place.

8.2 Population turnover and acceptance policy

Population turnover mechanisms were explained in paragraph 2.4.2, and were studied in the Naming Game in Steels and Kaplan, 1998 and Vogt and Coumans, 2003. An effect of population turnover is the instability of the lexicon: if agents are replaced faster than a certain rate, a reasonable level of agreement cannot be reached anymore. Active behaviors could help to change this critical rate to higher values.

It seems that another mechanism may play a more important role, potentially coupled with active topic choice: acceptance policy. Deciding whether to accept or not a new convention can be based on measures like LAPS, and older agents that have built enough confidence about the lexicon will not easily accept conventions invented by newcomers, when the latter will accept what they are taught.

An efficient setup to study this aspect would be to start with a population that has already converged, and keep track of the agreement (TCS measure) of the current lexicon present in the population with the initial lexicon. The agreement decays over time, and the decay period characterizes lexicon stability.

Preliminary work has been done, and shows that previous topic choice strategies are not efficient in this setting. An explanation would be that the mechanism used for level 2 of the topic choice policy uses a type of intrinsic reward that is not adapted to this case: agents should maybe keep reinforcing associations that they are the most certain about to at least keep a part of the lexicon stable, instead of the ones associated with most uncertainty.

The mentioned modifications have already been implemented in the framework: starting with a full lexicon, decay measures, and acceptance policies. New strategies for both acceptance policy and topic choice can be written and tested easily.

8.3 Diffusion on networks

As was explained in chapter 2, variants of the Naming Game include agents interacting not with everyone, but only with their neighbors on a *social network*. One of the issues encountered with the Naming Game is the diffusion of conventions on such networks: in some cases it can be slow, or even lead to the emergence of metastable clusters of agents having distinct lexicons. Active topic choice (and maybe acceptance policy, introduced in the previous section) may help solve these problems: if less conventions are created, the ones shared in each cluster may be complementary with each other. Also, the active mechanisms may help the selective diffusion of some meanings, and improve the overall diffusion time per meaning.

An interesting setup, apart from random networks, is a line of agents already sharing a completed lexicon for all meanings but one: the remaining meaning is associated to one word for the first half of the line, and another word in the other. This setup has been studied analytically and simulated for a single meaning in Baronchelli, 2006, and a diffusion coefficient can be defined. The effect of active topic choice on the diffusion coefficient could be studied.

Using networks between agents has already been implemented in the framework, along with the *half-line* setup.

8.4 *Structured meaning spaces*

In this thesis, we have only studied a finite meaning space without structure. It has been shown that intrinsically motivated behavior can use the characteristics of a structured meaning space to faster build a shared language (Cornudella Gaya, 2017), but could we adapt our topic choice policies to such cases?

8.4.1 *Continuous spaces*

A first example is the domain of color, a continuous space. A language game model in particular, called the Category Game, was introduced in Puglisi, Baronchelli, and Loreto, 2008. Agents see the color space as a one-dimensional segment, and simultaneously build a perceptual representation of colors by splitting this segment into subsegments, and negotiate a lexicon to be able to refer to those segments and subsegments. At the end of the simulation, the perceptual segmentation is as precise as possible, but the lexical segmentation regroups large number of segments. The average number of such categories matches more or less the number of colors observed in natural languages.

In this model, agents only have two possible meanings that they can refer during each interaction, sampled from a given distribution over the color segment. In this case, topic choice does not make much sense. Nevertheless, it is possible to sample several couples of colors in this way, each being a *context*.

Preliminary work showed that active context choice accelerates the dynamics, but leads to a lower number of lexical categories at the end: typically 3 or 4, compared to ≥ 10 in the original model. This is due to the inner mechanics of the model, but could be solved by considering a hierarchical version of the Category Game, introduced in Loreto, Mukherjee, and Tria, 2012.

The Category Game, along with a few associated measures, is already implemented in the framework. However, it is not computationally efficient.

8.4.2 *Zipfian bias*

Another type of structure is a prior bias when sampling on meaning and/or word spaces. In fact, word occurences in speech are not uniform, but follow a powerlaw distribution called Zipf's law (Zipf, 1949). Also, it is obvious that some meanings are relatively more important or useful than others: eating and sleeping are naturally more frequent in language than PhD thesis or *success threshold*. However, in certain contexts, their relative utility might change, and as a result so would their relative frequencies.

Random sampling on meaning and/or word spaces could be changed from uniform to zipfianly biased. This can help to limit the complexity burst (few meanings are concerned by the first inventions, and conflicts are solved faster), but also slow down the convergence process: some meanings are rarely selected. Using active topic choice, it is possible to take advantage of the first feature (when meanings are considered equivalent), while not having the drawback of selecting some meanings only with a low probability.

Zipfian biased meaning and word spaces are already implemented in the framework.

8.4.3 Expanding spaces

We have seen an example of non-finite meaning space with the Category Game, but this case was a continuous space. An unbounded but discrete meaning space cannot be accessible from the beginning to an agent: they would never talk twice about the same meaning if using random topic choice. An alternative is to build accessibility to this space meaning per meaning: starting from a core meaning or group of meanings accessible to every agent, the accessible space would expand when some of the accessible meanings are chosen as topic. Meanings can be seen as nodes of a graph, and talking about one triggers accessibility to its immediate neighbors. The accessible meanings that have not been explored, at the edge of the accessible graph, constitute the *adjacent possible*, a concept introduced by Kauffman, 1996 to refer to the constant expansion of the space of possibilities due to the occurence of novel events.

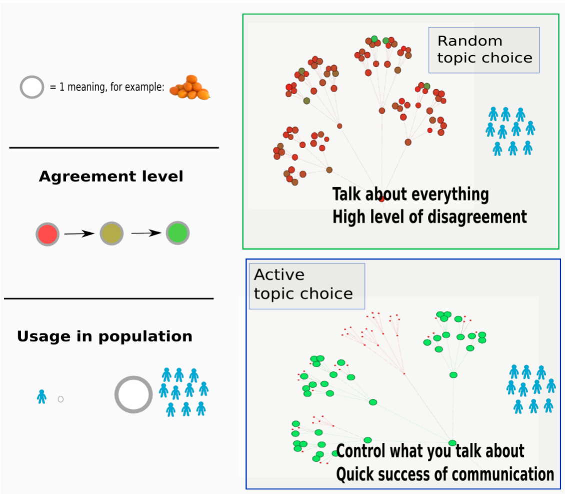


Figure 8.1: Illustration of the dynamics on a meaning space structured as a graph.

Preliminary work shows that Random Topic Choice explores faster, but without reaching an agreement on individual meanings. Active topic choice strategies can limit exploration so as to reach full agreement, and the number of explored meanings grows linearly. An interesting direction in this model is to see if Zipf-like distribution would arise from meaning usage. In fact, it has been shown than zipfian distributions are related to this kind of exploration/innovation processes (Tria, Loreto, and V. Servedio, 2018).

The graph structure of the meaning space and the exploration dynamics are already implemented in the framework. A first visualization of the dynamics of random vs. active topic choice can be seen on figure 8.1 or in video here:

https://www.dropbox.com/s/bl4ytaykbmcnlta/ngal_struct.mp4?dl=0

8.5 Consensus dynamics

Communication between individual agents without a centralized control is not only found in humans, but has a role in many computational problems: coordinating the movement of a swarm of drones, routing data on a big network like Internet, synchronizing parallel tasks on nodes of a computing cluster,... In particular, it can be necessary to reach a consensus on the value of a variable that depends on the state of all agents, as for example the maximum temperature of a node on a computing cluster, the average humidity of a zone for drones doing meteorological monitoring, or which hubs are more reliable on a network. Those problems are gathered under the term *consensus dynamics* (Ren, Beard, and Atkins, 2005), and are studied in various conditions with constraints on memory, network structure, and pre-shared knowledge by the agents. These problems are related to our models, even if the interaction scenario of our model is very specific and the end consensus cannot be known in advance. Typically, algorithms in this field are designed and studied to solve one consensus problem, but our algorithms could be helpful when there are several parallel consensus to be agreed upon simultaneously.

8.6 Human behavior

8.6.1 User experiments

As discussed in chapter 6, several possibilities are offered by the framework that we built to conduct our experiment, including bigger meaning spaces and changing ending conditions, to study both exploratory behavior and memory limitations. Also, all modifications to the standard Naming Game model can be directly imported and studied as a user experiment.

A particular alternative direction for the user experiment is the multiplayer version introduced in the discussion of chapter 6, and already used in previous scientific work (Steels, 2015) or artistic exhibitions[1]. In this setup, each participant has its own population of simulated agents, and exchanges agents with a global pool. Agents remember what they learned with one participant, and will spread their knowledge when interacting with others. In this way, no synchronization is needed between participants, and there is no observed latency for players. Agents could also adapt their strategies

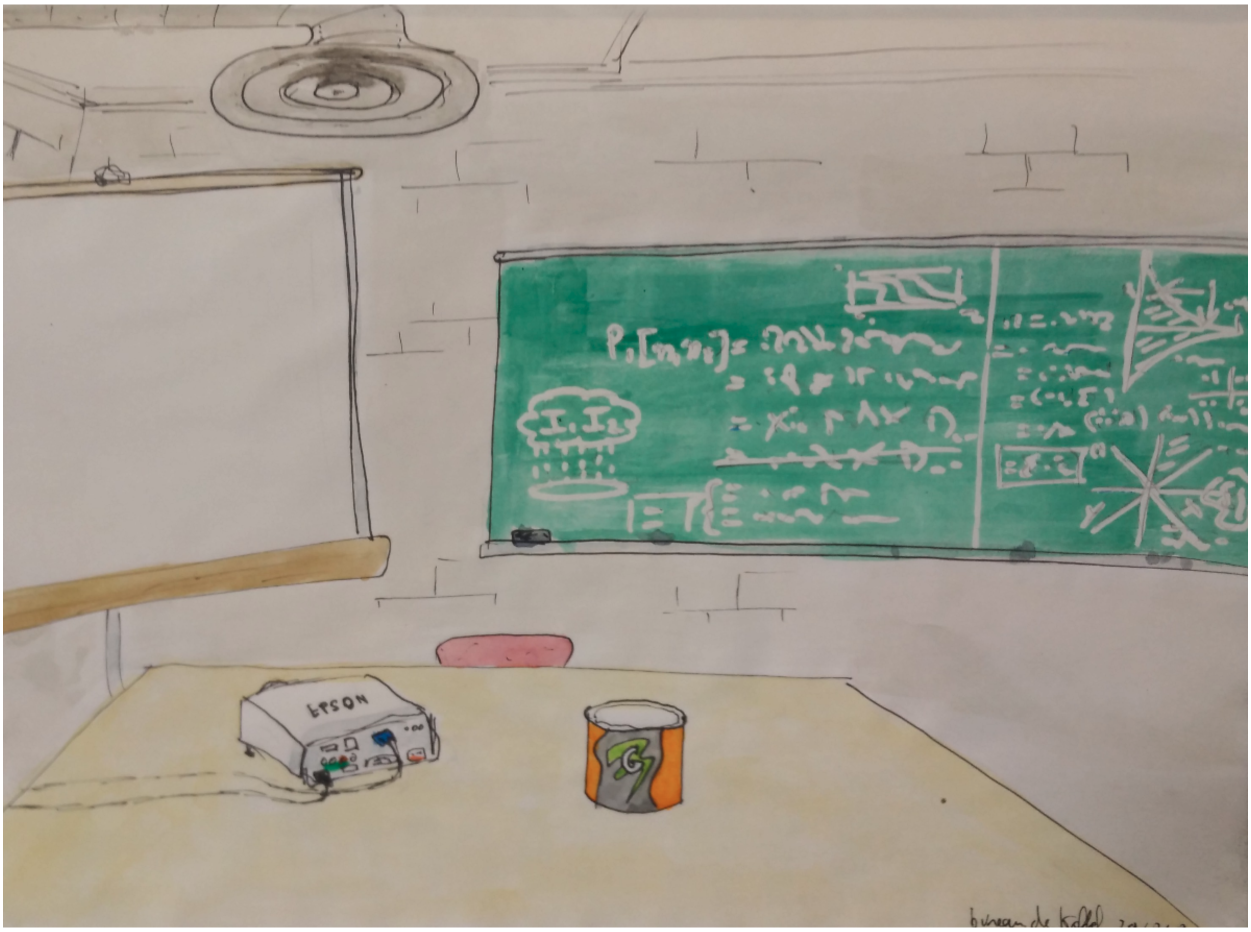
[1] *Mathématiques, un dépaysement soudain*, Paris, 2011

to mimick the player's, thus being closer to a Naming Game with only human agents. This version of the experiment is more prone to biases and somehow ill-defined situations, but it could make the experiment more attractive for participants, and reach a larger population on a longer term.

8.6.2 *Analyzing existing databases*

If recruiting participants for user experiments can be complicated, there is another way to collect behavioral traces: use existing databases. Linguistic properties have for example been studied Twitter data (Mocanu et al., 2013), and predictions of Language Games models were compared with databases including a survey on statistical properties of color names in natural languages (Kay et al., 2009) for the Category Game (Baronchelli, Gong, et al., 2010; Puglisi, Baronchelli, and Loreto, 2008), or census data from the United States to model the emergence of creole languages (Tria, V. D. P. Servedio, et al., 2015).

Recently, work has been done on the analysis of online corpuses and find a pattern of topic change (Karjus et al., 2018) but on the time scale of several years. In Grieve, Nini, and Guo, 2018, a method is defined to identify lexical innovation and spreading, using Twitter data (i.e. on much shorter time scales). A combination of both approaches could be used to identify topic choice policies. However, the dynamics and importance of topic choice on expanding spaces would have to be studied first, as the meaning space is unbounded when considering this kind of data.



Useful conversations with T. Koffel, Kellogg Biological Station, Michigan. July 2018

Appendix A
An open-source simulation framework

Contents

In this appendix, we will present two Python libraries that we built in the last few years, and how to rerun the experiments of this thesis using them. The two libraries are:

NamingGamesAL: This library gathers all modular parts of the Naming Game and its variants.

Experiment manager: This library generalizes the concept of running multiple computational models by abstracting away the execution, update and parameter management parts.

They can be found respectively on:
http://github.com/flowersteam/naminggamesal
http://github.com/wschuell/experiment_manager

In this appendix, we will not spell out the complete URLs to refer to specific files but their position in the tree of the corresponding repository. Links are still clickable.

It has to be mentionned that those libraries were developped under the course of the PhD, and are not well commented and documented yet. They are still intertwined for a few parts, but will be clearly defined separately in the near future. This appendix is a glimpse at the possibilities offered by those libraries, and a few guidelines to start using it. Reproducing the simulation results presented in this thesis is easily doable thanks to the configuration files that are to be found on another repository:

http://github.com/wschuell/notebooks_thesis

A.1 Simulating the Naming Game and its variants

The first Python library, Naming Games AL (for Active Learning) gathers a series of modular elements to simulate the Naming Game. Configuration for simulation are a nested dictionary describing the different modules. Below is an example of configuration for a population, and how to use it to create the Population object:

```
import naminggamesal as ngal

pop_cfg = {
    "nbagent": 5,
    "strat_cfg": {
        "vu_cfg": {"vu_type": "minimal"},
        "success_cfg": {"success_type": "global_norandom"},
        "wordchoice_cfg": {"wordchoice_type": "random"},
        "strat_type": "naive"
        },
    "env_cfg": {
        "env_type": "simple",
        "M": 5,
        "W": 20},
    "interact_cfg": {"interact_type": "speakerschoice"}
    }

pop = ngal.ngpop.Population(**pop_cfg)
```

The configuration mirrors the hierarchical classification that was introduced in chapter 2. It describes here a population of 5 agents, in a world with 20 words and 5 meanings, using Speaker's Choice scenario, Random Topic Choice (*naive* strategy) and random word choice. An more complete view can be seen in figure A.1. The different values possible for each module can be seen in the corresponding folder. For example for Word Choice Policy, it is situated in the file ngstrat/word_choice/__init__.py.

To run the simulation for one interaction:

```
pop.play_game()
```

Population are encapsulated in *Experiment* objects, linked to a local database. Their purpose and structure will be described in the next section, but we can already show how to use them. In the code below, an experiment will be instanciated with the population configuration defined above and run for 100 interactions. We will then compute and plot $S(t)$, the theoretical communicative success. This measure will is noted *srtheo* for *success rate theoretical*.

```
db = ngal.ngdb.NamingGamesDB()
xp = db.get_experiment(pop_cfg=pop_cfg)
xp.continue_exp_until(T=100)
xp.graph('srtheo').show()
```

A.2 *Experiment manager: the need for a simulation framework*

In this section, we will briefly present the experiment manager framework. A much more detailed description could be provided,

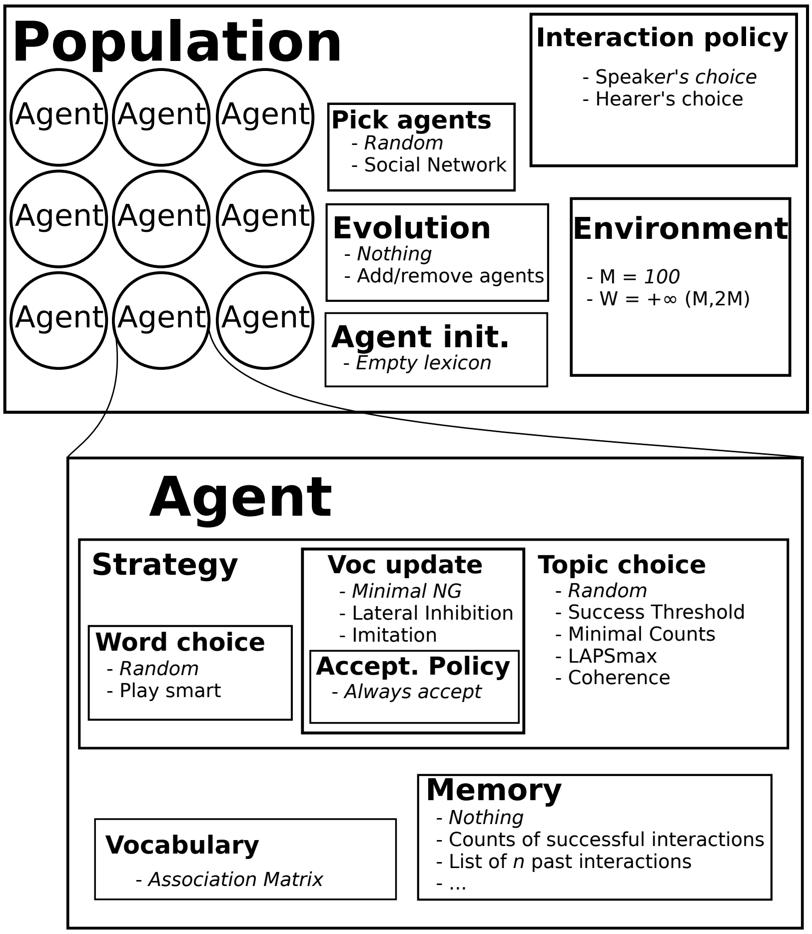


Figure A.1: Modular representation of the NG models used in the library. Some modules are not represented, but these are enough to understand all experiments of the thesis.

but is beyond the scope of this thesis. We will focus on introducing the key concepts and commands to be able to rerun and analyze the experiments of the thesis.

If the library is still a bit entangled with the Naming Games AL library, we are slowly converging towards a generalized framework.

Motivation and basic concepts

The behavior of many models and dynamical systems can be difficult to describe analytically. However, computer simulations are an efficient tool to gather data and empirical evidence, giving powerful clues and hypotheses that can sometimes lead to theoretical findings. This approach can be used in many scientific fields, not only physics but also ecology and social sciences. Models often rely on many parameters, and searching for interesting behaviors can result in the necessity of conducting big experimental campaigns, running simulations for a lot of different parameter settings. Keeping track of the data, the parameters, and controlling execution (more efficient on a computing cluster for example) is usually done by the experimenter, and may be at the limit of the scope of their skills. Also, managing this part of the code can induce bugs and errors, and therefore constitute an unnecessary weak point in the scientist's work.

All those simulations have a structure in common: an object (the dynamical system) goes through a series of steps, its evolution at every step being defined by a given set of rules. Metrics can be monitored over time, or at the final step. The construction of the object uses the values of the parameters, and can often be organized in a hierarchical way, as for the Population object that was introduced in the previous section. If the construction and rules evolution of the object are specific to the model, the remaining structure of the code is usually really similar from model to model. However, many scientists spend a lot of time to recode those parts, and each time introduce new bugs, or copy/paste existing ones and fail to efficiently adapt old code to new models.

With the Experiment Manager library, our aim is to create a framework and specific concepts to generalize this part of the simulation code, and make it reusable for new models.

Our starting point will be the dynamical system, with a set of rules to construct it or *configuration*[1], the rules of its evolution, and a list of metrics on this object.

[1] containing among other things the parameter values

Experiment, batch- and meta-experiment objects

Experiments are the basic concept of our framework. An experiment encapsulates an object (the dynamical system) and manages its evolution and monitoring. Experiments are linked to a database (typically SQLite), where they are stored alongside their associated monitored values. They are not defined by the values of the parameters that were used to construct their object, but have a unique ID (UUID). In fact, models are often stochastic, and the evolution of the dynamical system can be different even if the object was the same at the beginning. Conducting several experiments with the same configuration can be necessary to estimate an average behavior of the object. With a UUID, it is possible to have several experiments with the same configuration and keep track of them.

Experiments also have a random seed, defined at creation, to make their execution deterministic, and interruptable.

If the experiments run for a high number of steps, we may not need to record the monitored metrics for all steps: it could induce a high memory usage, as well as an unnecessary computational overhead. Experiments can be specified a step policy, for when to compute those metrics. By default, we chose a near-logarithmic evolution of the monitored steps; keeping round values: from 10 to 40, every value. From 40 to 70, every second step. From 70 to 100, every 5 step. Then, start over at the new order of magnitude. This entails a logarithmic computational cost of the monitoring metrics with the maximum step T_{max} instead of linear.

Experiments can either store *snapshots* of the simulated object to be able to compute monitoring functions later, or compute them while running the experiment and only storing the last snapshot, or current state of the object.

Batch-experiments regroup many experiments, potentially with different configurations, and manage their construction and execution. They package the experiments as *Jobs* and executes them through a *Job queue*, two concepts that will be introduced in a paragraph further in this section. Batch-experiments also take care of defining what are the metrics to be monitored on which experiment.

Meta-experiments are the object that we will mostly interact with: they correspond to the level of an experimental campaign. They possess a batch-experiment object, and use it to run their associated experiments. But the main feature of meta-experiments is parameter management: they are provided a list of parameters, each with a name, a range, a label, and a default value. The meta-experiment also has two functions: one to build the configuration from the parameter values, another to define the maximum number of steps. Using this information, experiments are built and executed through the batch-experiment. Plotting metrics is straight forward, using this type of syntax:

```
meta_exp.plot('srtheo')
meta_exp.plot('srtheo',N=10,nb_iter=10)
meta_exp.plot('srtheo',N=[10,20,50],nb_iter=10)
meta_exp.plot('srtheo',N='all',M=100,nb_iter=1)
```

Correct labels, units, minimum and maximum values, and even legend when several configurations are asked are automatically set. A default number of iterations is associated to the meta-experiment, but can be changed while using the plot instruction. When missing from the command, a parameter is given its default value. Metrics that have a value only at the end of the experiment (global metrics), can be plotted with respect to a parameter:

```
meta_exp.plot(measure='conv_time',token='N')
meta_exp.plot(measure='conv_time',token='N',M=100,N=[10,20,50])
```

Examples of usage can be seen here: github.com/wschuell/notebooks_cogsci2018

Execution: Jobs and Job queues

We have talked so far about parameter management and keeping track of individual experiments, but not of how and where they are executed.

Jobs are packaging experiments to be executed: isolating them in a specific folder, keeping track of execution time, saving checkpoints when necessary and keeping a backup copy of all associated files as well as checksums to verify integrity. Another important feature of jobs is the possibility to profile the execution, helping to detect particularly time consuming parts of the code. Typically for a meta-experiment, the last possible configuration (often associated to the highest values of parameters and therefore the longtest to execute) is used to set up a profiled Job with a maximum number of steps divided by 100. It will run way faster than the normal configuration, and provide profiling information. The notion of jobs is not specific to experiments: other subclasses of Jobs have been implemented, as examples or templates to package non-Python code.

Job Queues ensure that Jobs are executed where they should be: locally, locally using all processors available, on a remote server, or on a computing cluster. When used with a computing cluster, Job queues can be provided the specific policies of that cluster: syntax to talk with the scheduler, folders to do the execution, maximum number and time for job submissions, and efficient data transfer (regrouping files when needed, as file by file can be really long over SSH). When jobs are submitted to the cluster with an estimated execution time that is too low, they are resubmitted in their current state (last checkpoint) with a longer time. Job queues can keep track of average completion level of jobs. Also, Job queues ensure that the last version of your code is on the cluster, by setting up python virtual environments and for example pulling the last version of the code from an online git repository. After execution, Jobs are retrieved and unpacked: experiments are reintegrated in the local database.

All these features are completely transparent to the user, provided they set up correctly the meta-experiment. The creation and execution of all experiments can be done with the sole `meta_exp.run()`. Changing the default job queue is not more complicated: `meta_exp.run(batch='local_multiprocess')`.

A.3 ReScience: experiments of this thesis

Procedure to rerun experiments

Dependencies: both libraries (naminggamesal and experiment manager). If there are troubles to install them, a docker-compose file is present at the root of the thesis repository.

For the thesis repository (github.com/wschuell/notebooks_thesis), we do not rely on a single meta-experiment, but on a few dozens of them. We designed an even higher level, where meta-experiment are automatically generated, using a set of possible configurations and parameter definitions. On the repository, `configs_gen.py` contains the list of meta-experiments, and relies on configuration files in the `configs` folder. Running this python file creates each folder containing a `metaexp_settings.py` file. Running it runs all the associated experiments, and later importing it enables to plot and analyze data, for example in an interactive notebook. In particular, the `configs` folder contains a `metrics.json` file describing all available metrics.

An important remark: execution directives may need to be changed to rerun some experiments to local execution. By default, we were using the *Avakas* computing cluster.

Use databases

Because rerunning all experiments can be extremely long: we estimate at many months, probably years, the CPU time that was used to generate the data. Databases for all meta-experiments will be made available, as a link on the repository description. The total size is a few gigabytes.

Replotting figures

Running the `configs_gen.py` file will generate a `get_figs.py` file in each folder corresponding to a meta-experiment. Executing the latter file will recreate the figures. They are all defined in a monolithic file, `get_figs_list.py`, that is not executable but is parsed to generate the individual python files. The content of a `get_figs.py` file can be for example transferred in an interactive notebook and reused/modified to plot similar figures.

CogSci Conference, Madison, WI. July 2018

Appendix B
User experiment web application

Contents

The user experiment was built and deployed as a web application, written in the Django Python framework. The experiment is freely accessible at http://naming-game.space.

B.1 Setting up, testing and running analysis on a new server

Docker containers are a way of packaging software so that it can run easily on very different computer systems, without having to worry of dependencies and compatibility. Moreover, containers packaging different softwares (a database, a specific server, etc.) can be combined with a tool called *docker-compose*.

We use this framework for our server, composing three containers: a *PostgreSQL* database, an *nginx* web server, and a container specifically for our application, using written in the *Django* Python framework. The only dependencies are *Docker*, *docker-compose* and Python (docker-compose requires Python). Instructions to install them can be found here. The few commands that have to be executed outside Docker suppose that you have access to a Unix shell.

The first step is to clone the git repository:

```
git clone http://github.com/wschuell/ng_userxp
```

Then, change directory and simply run:

```
cd ng_userxp
docker-compose up
```

The docker containers should be installed, up and running in about 10 to 30 minutes, depending on your bandwidth and system. Of course, a later execution will only require half a minute to be operational. To connect to the application, if you are running the service on your local computer visit the following URL in your browser:

http://127.0.0.1 If you are running it on a distant server, change the IP by your server's IP or name. To avoid killing the process when disconnecting from teh server or closing the terminal, you can run it as a daemon with the option `-d` or run it in a detachable terminal like `tmux`.

To run it in Django debug mode, here is the command:

```
docker-compose -f docker-compose.yml -f docker-compose.dev.yml up
```

Update: updates are done via pulling from the source git repository. In parallel the library NamingGamesAL can also be updated, by default from the branch *ng_userxp*. A script at the root of the repository does it all, and can be executed while the server is running: `bash docker_update.sh`

Administration: The Django administration app is available on http://127.0.0.1/admin. The admin login and password are randomly generated at the first run of the server, in the file *superusers.json*, at the root of the repository.

Load testing: To test connections of multiple fake users on the server, you can run `locust --host=http://0.0.0.0`. Then connect to http://127.0.0.1:8089 to manage the tests. After testing, fake entries in the database can be removed by running `bash remove_locust.sh`.

Data backup: At each update, a backup of the PostgreSQL database is done and compressed, named by date in the folder *backup_postgres*. However, this cannot be used directly for the analysis. To get the json file usable for the analysis, run:

```
docker exec -it ng /bin/bash -c "python3 manage.py dumpdata > dump_tests.json"
```

This will create the file *dump_tests.json*, that you can use for the data analysis.

Data analysis: Place the json file obtained earlier in the analysis folder. In this folder, run `python3 analysis.py`. Tools to manipulate the data can be found in this file and in `measures.py`.

Changing the behavior of simulated agents: as well as other parameters of the experiment, can be done in the file ng/models.py. The NamingGamesAL configuration dict for example is located at the beginning of the file.

B.2 Screens of the first version

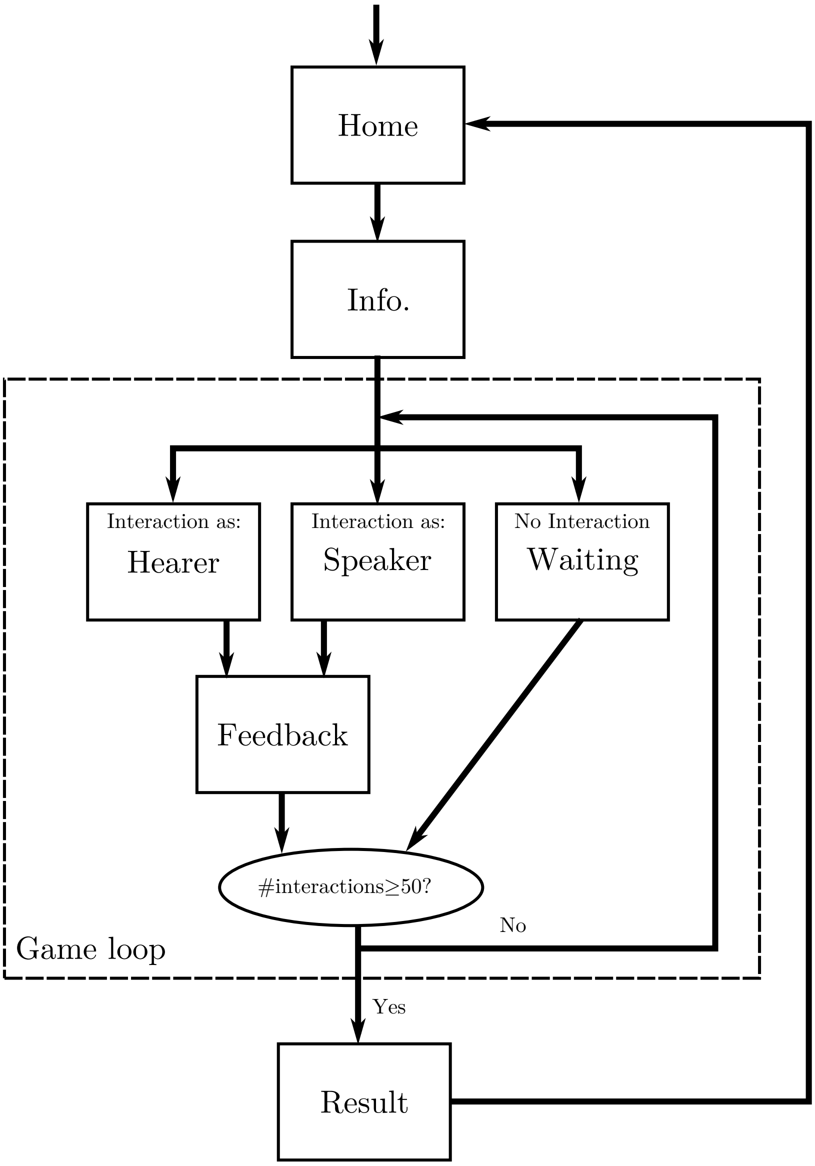


Figure B.1: Reminder of the structure of the user experiment: the different screens are represented by black rectangles, and the transitions between them by arrows. The participant arrives on a *Home* screen, showing some information and a button to start playing, leading to an *Information* screen, explaning a few details concerning the context of the game. The rest of the game is a loop: for some interactions the user is not involved, indicated on a screen *Waiting*; for the others they are either *Speaker* or *Hearer*. As a speaker, the participant chooses a meaning and a word. As a hearer, a word is provided, and the participant picks a corresponding meaning. After the respective choices, both screens lead to a *Feedback* screen telling them the result of the interaction: Failure or Success. If a count of $T_{max} = 50$ interactions has been reached, the game stops and a *Results* screen shows the final score (level of agreement, total and per meaning). If the count has not been reached, a new interaction starts and the next screen is again one of the three possible roles: *Speaker*, *Hearer* or *Waiting*.

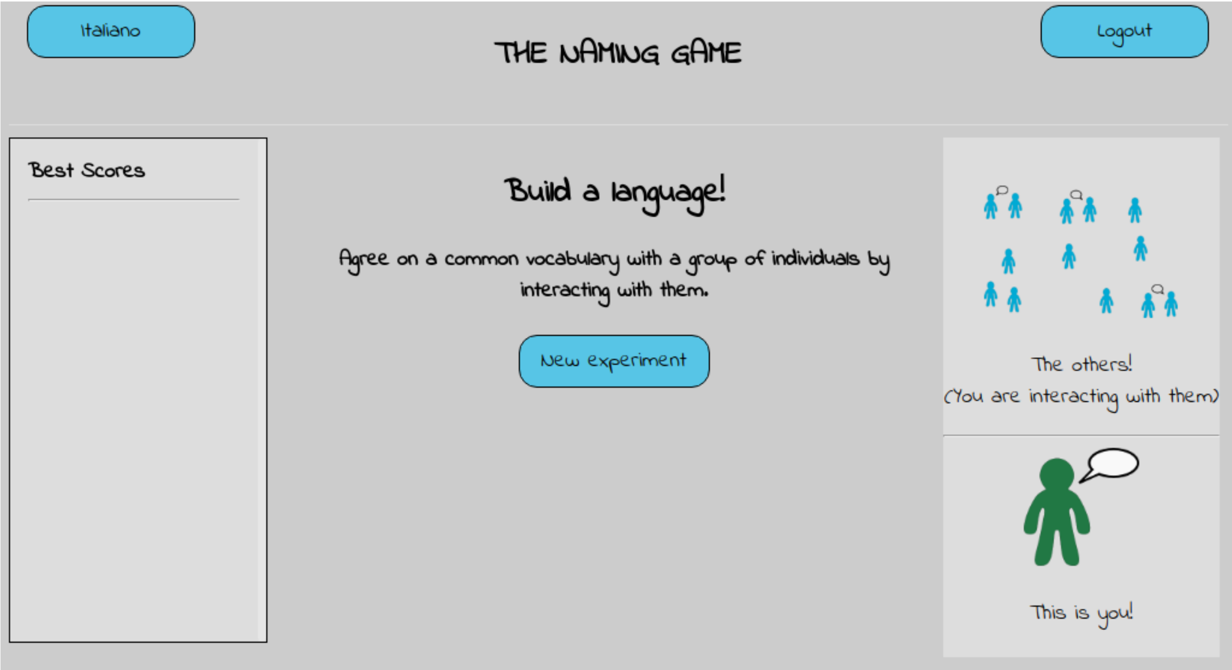


Figure B.2: Home

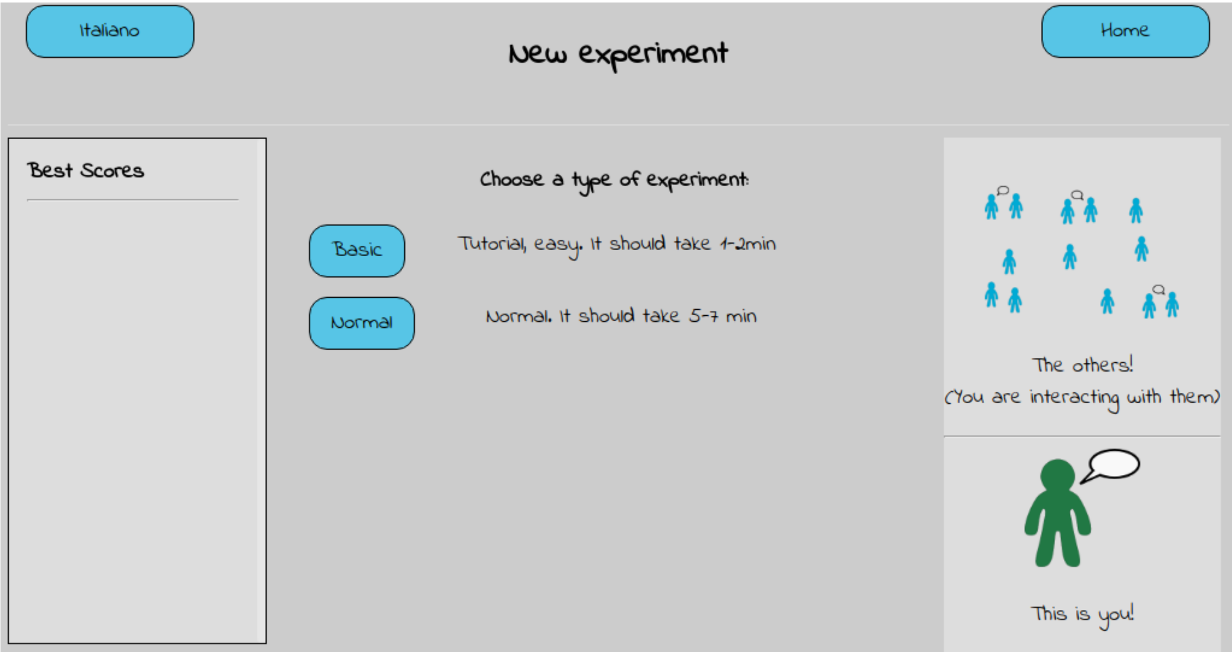


Figure B.3: Info screen: page 1/2

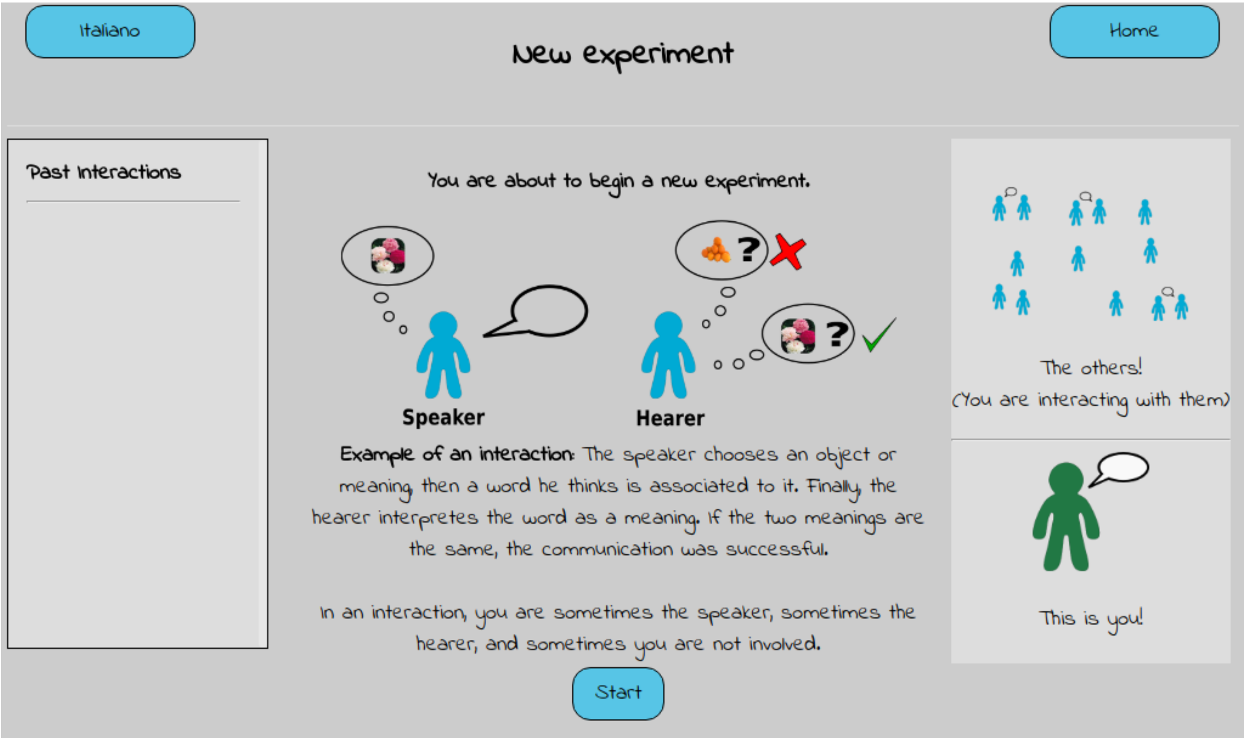


Figure B.4: Info screen: page 2/2

Figure B.5: Waiting

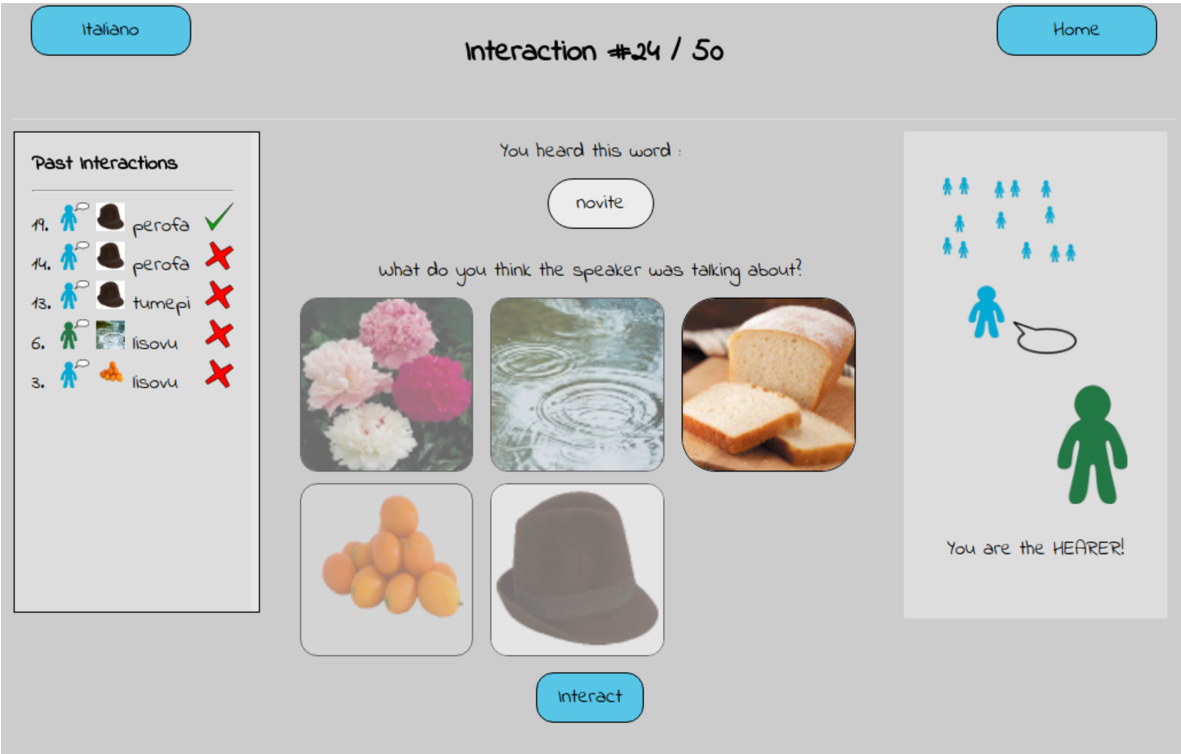


Figure B.6: Hearer

Figure B.7: Speaker

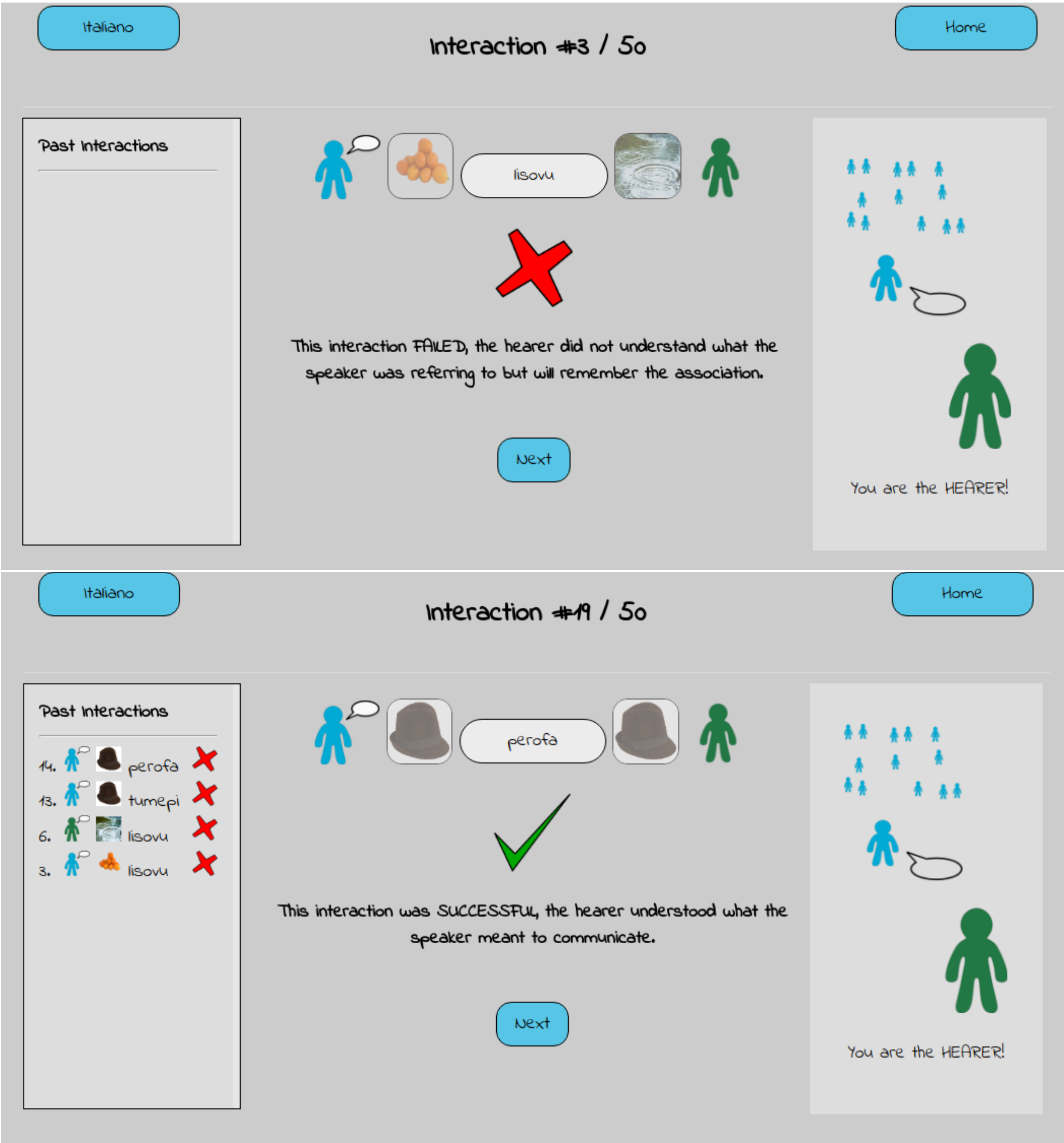


Figure B.8: Feedback screen: failure or success

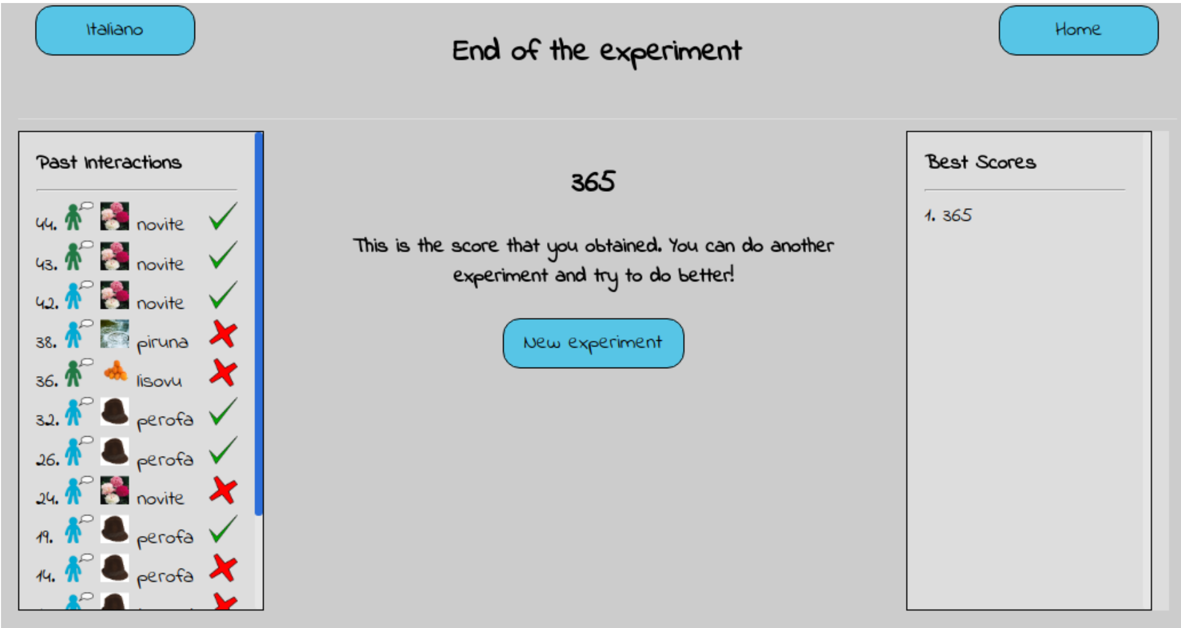


Figure B.9: Result

B.3 Screens of the second version

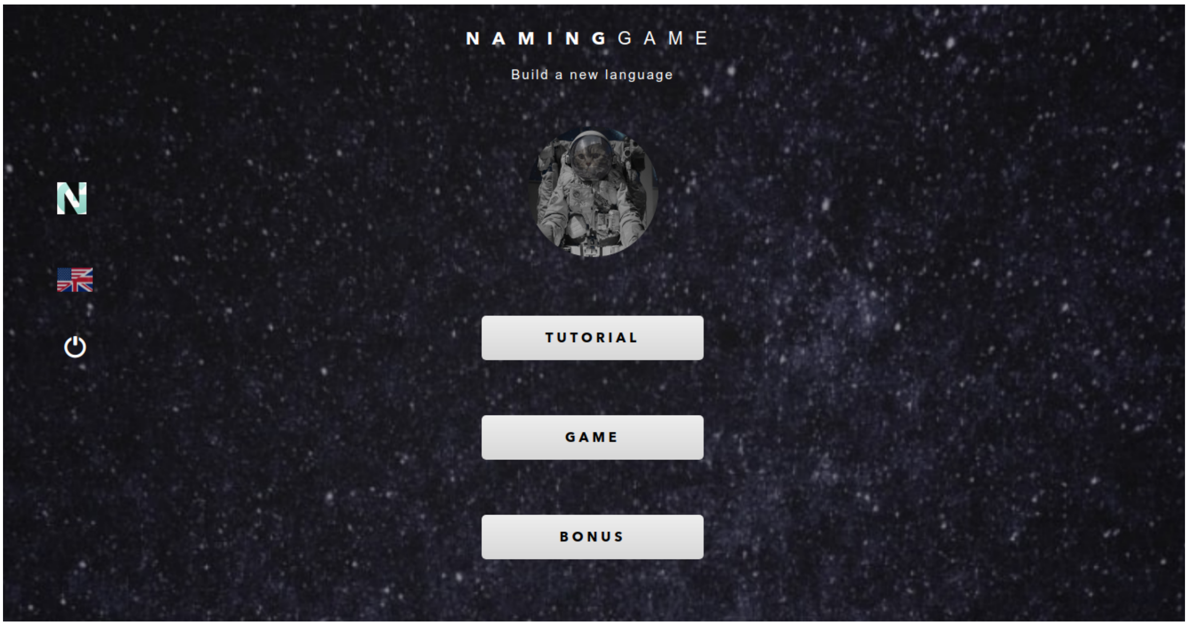


Figure B.10: Home

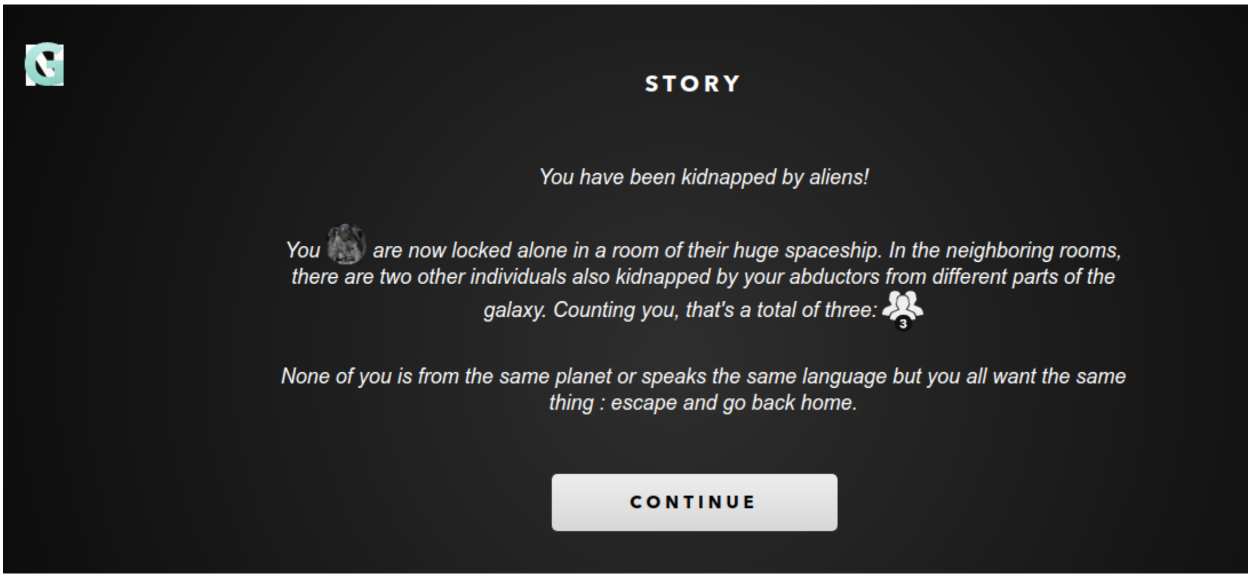


Figure B.11: Info: page 1/2

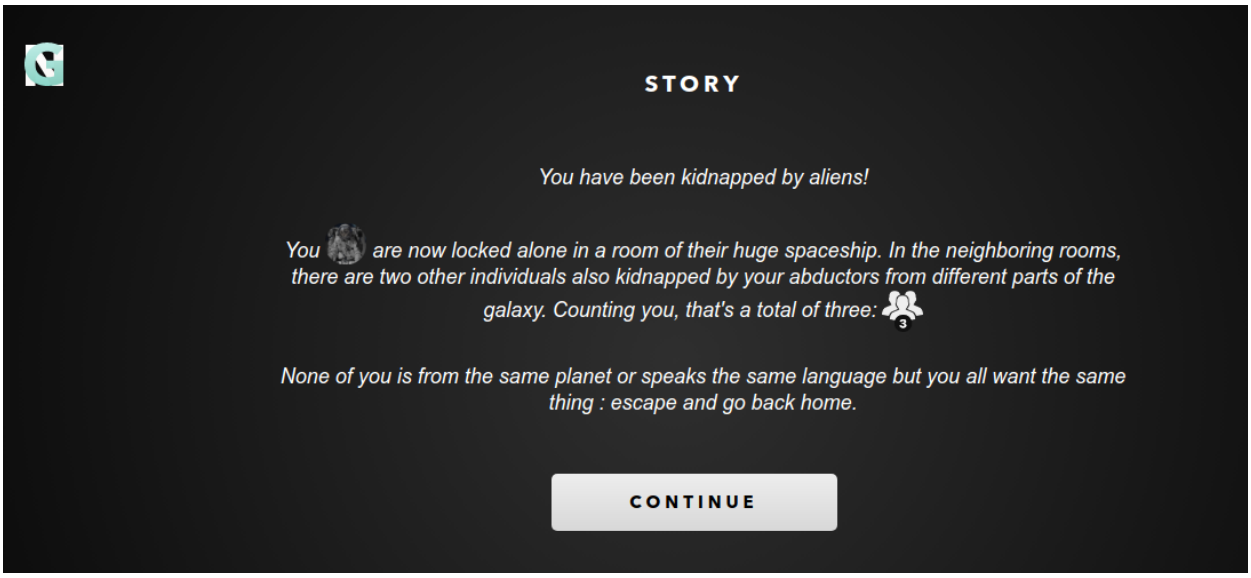


Figure B.12: Info: page 2/2

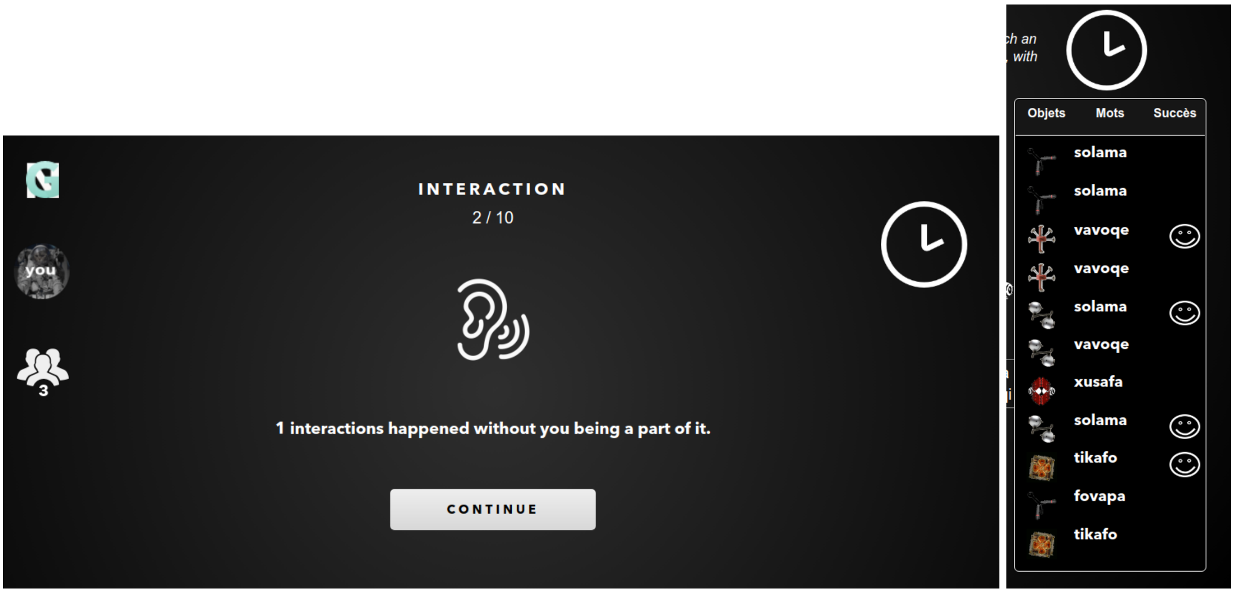


Figure B.13: Waiting, and past interactions information

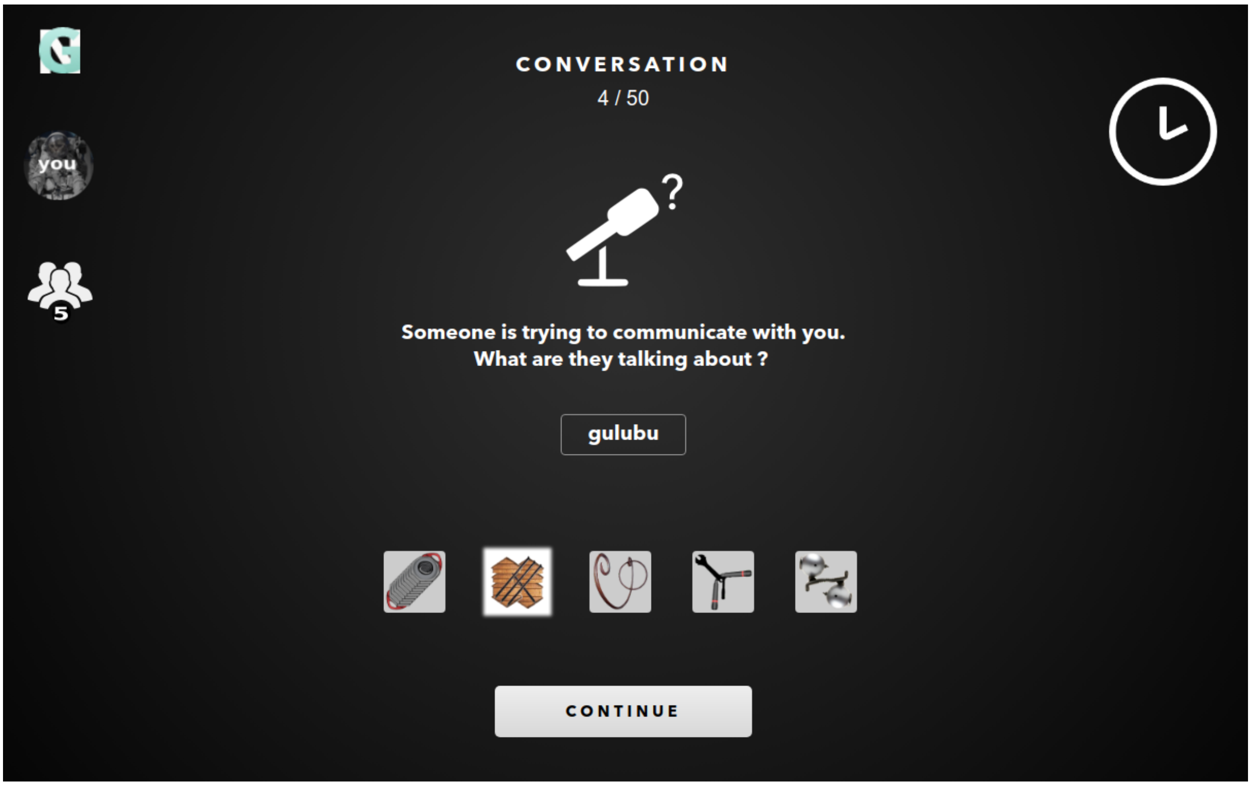


Figure B.14: Hearer

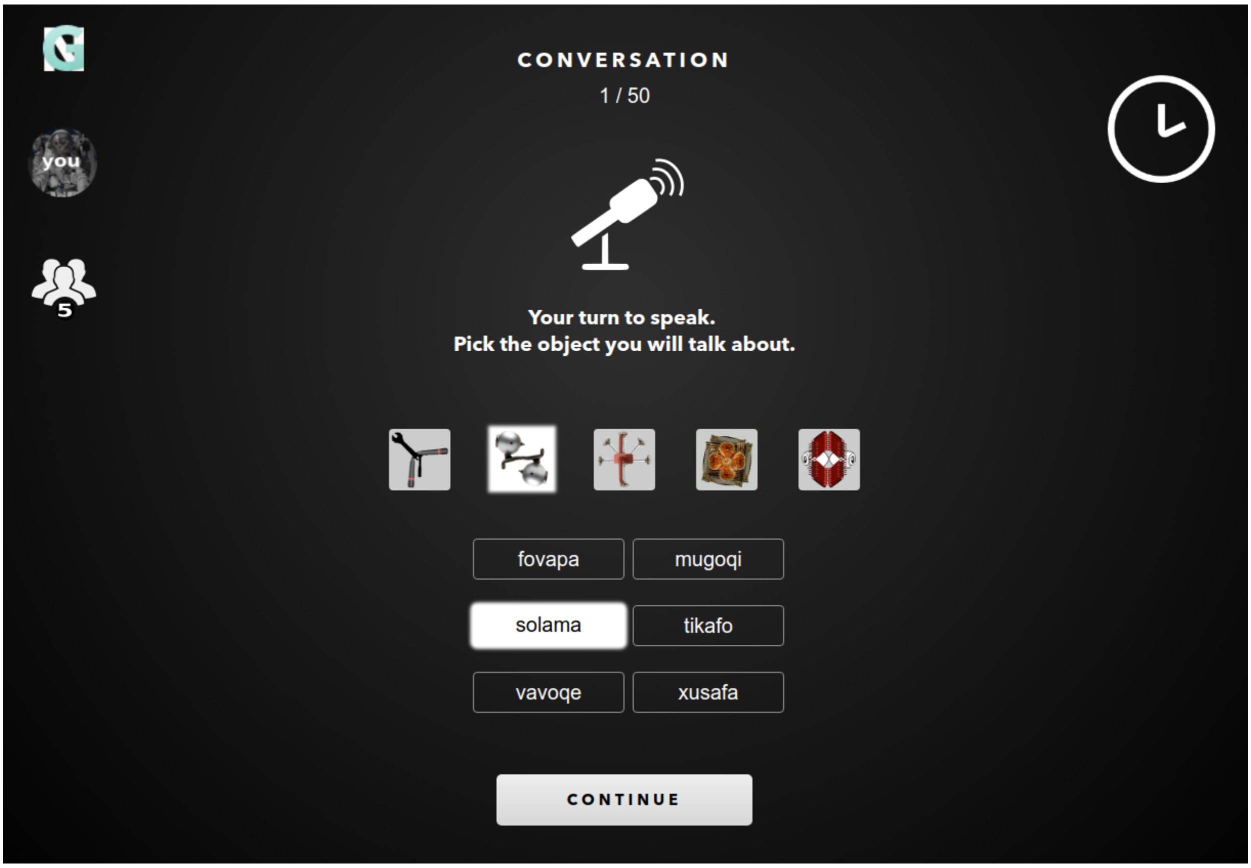


Figure B.15: Speaker

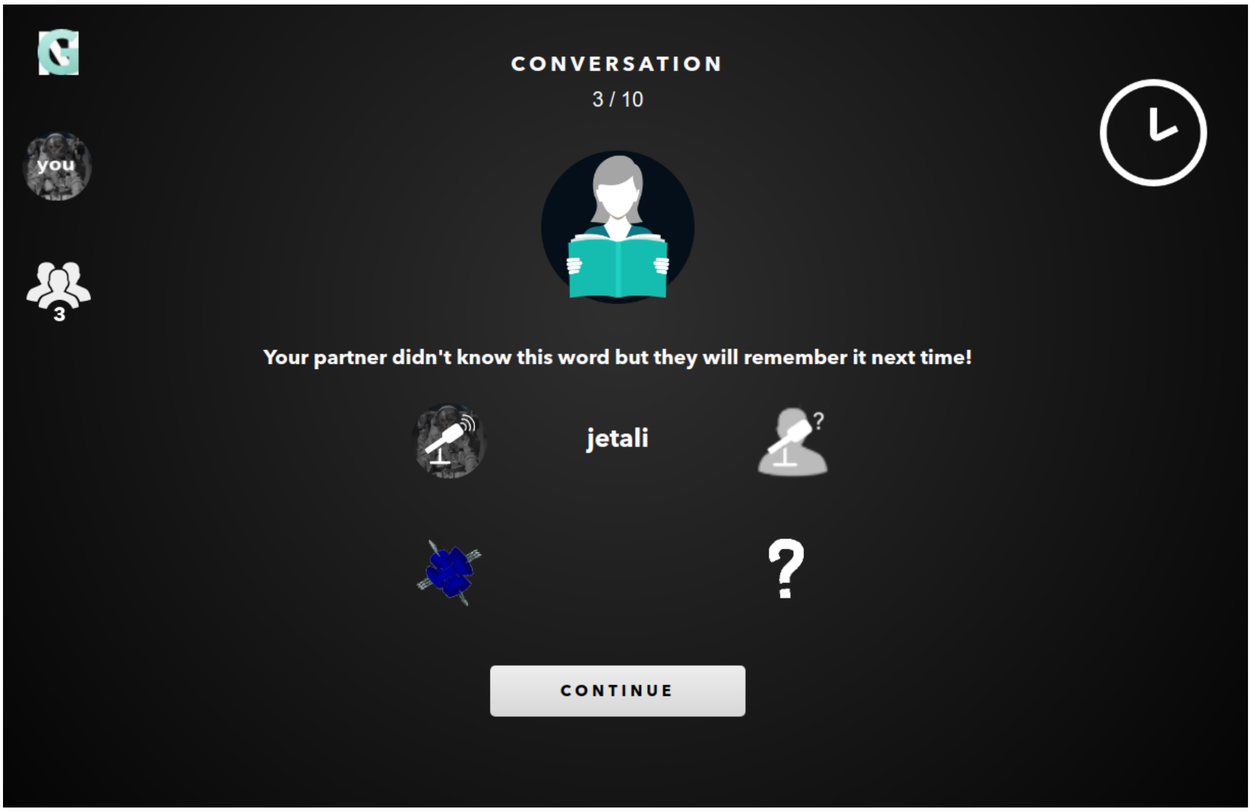


Figure B.16: Feedback screen: learning

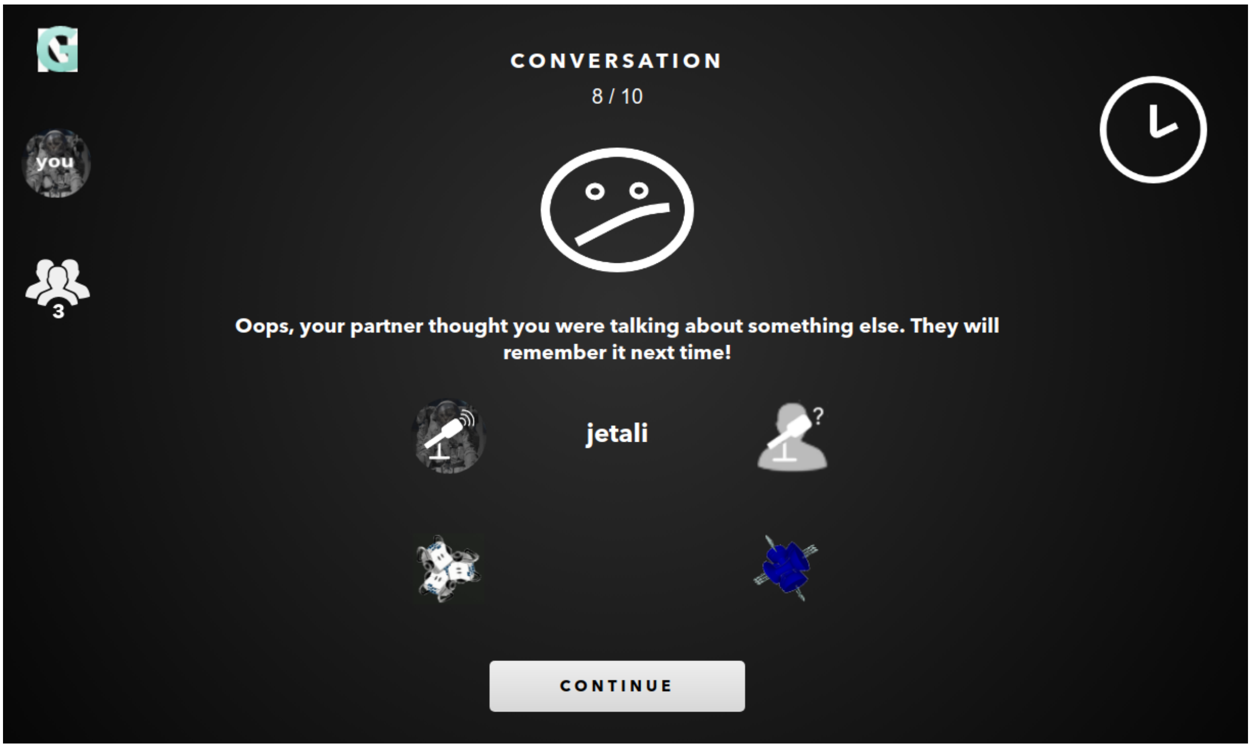


Figure B.17: Feedback screen: failure

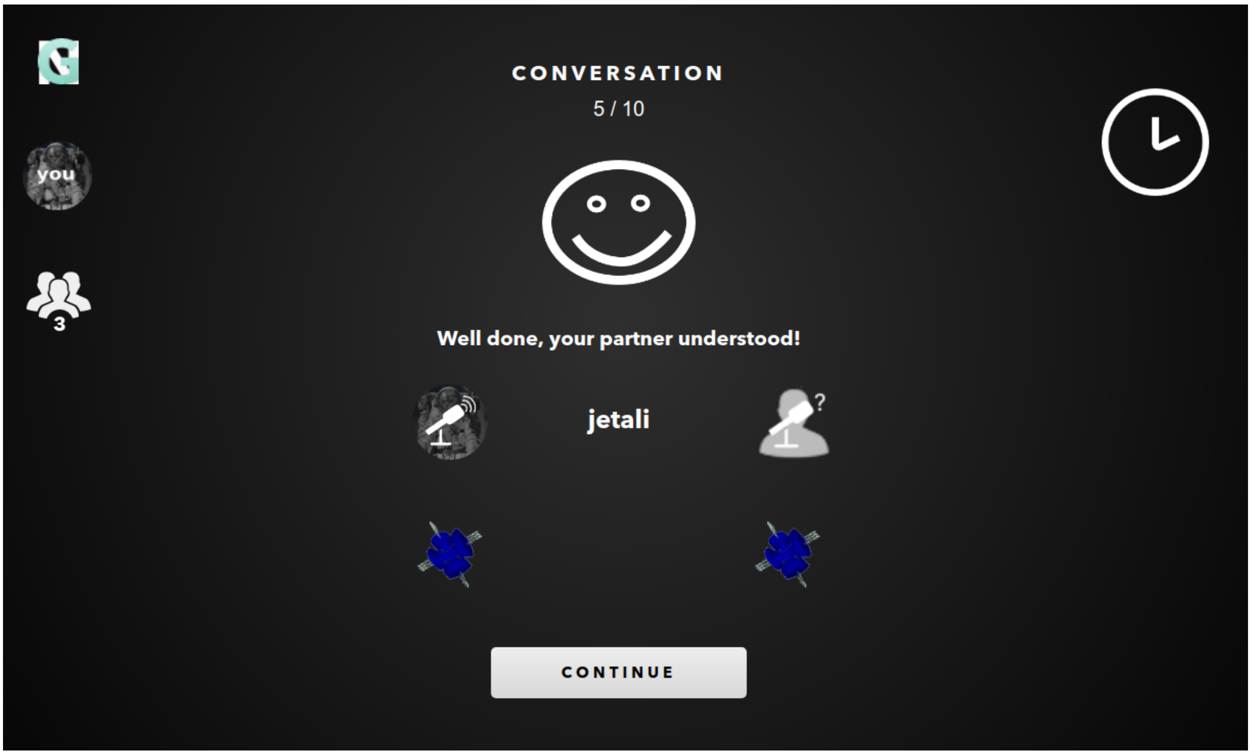


Figure B.18: Feedback screen: success

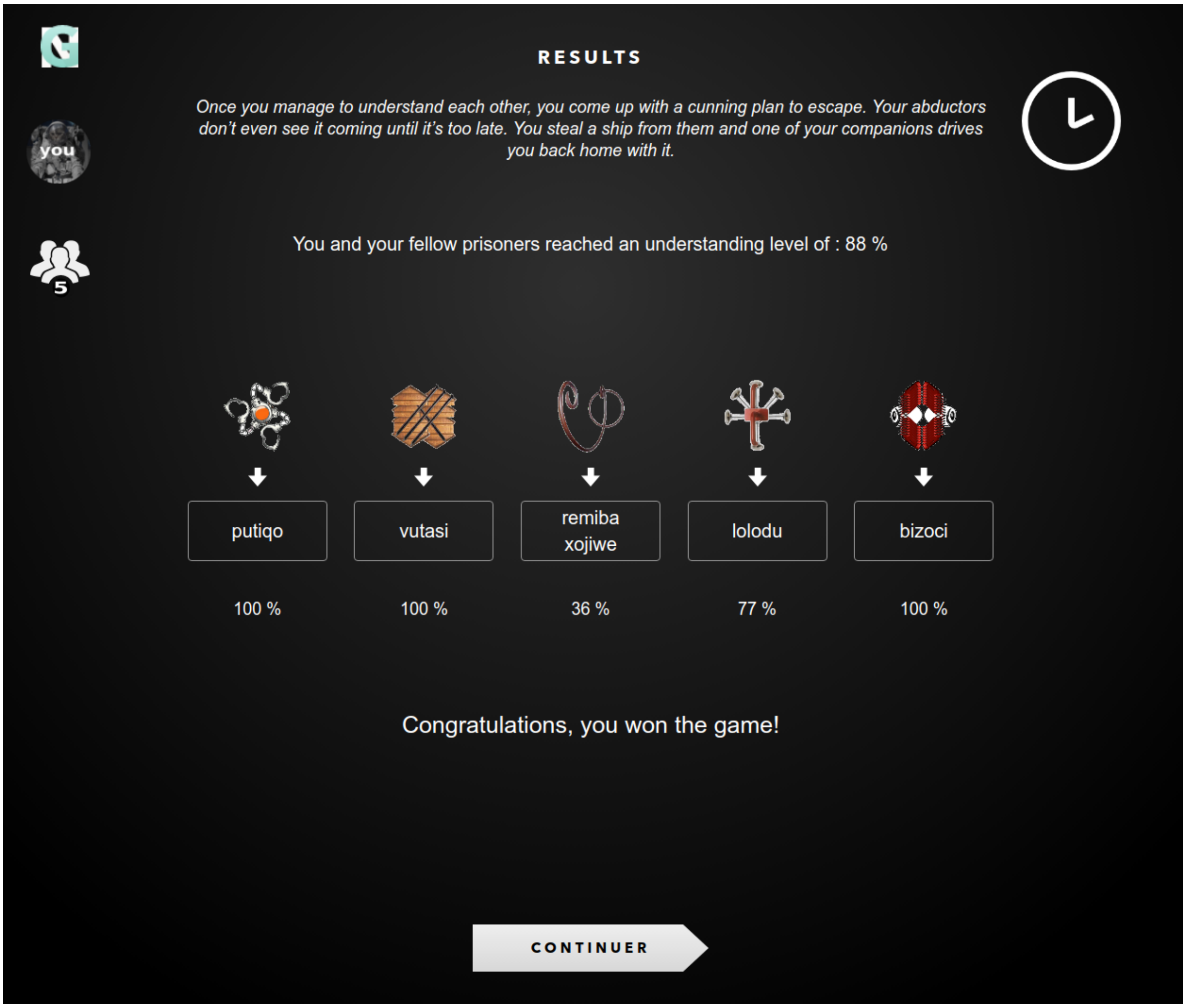


Figure B.19: Result

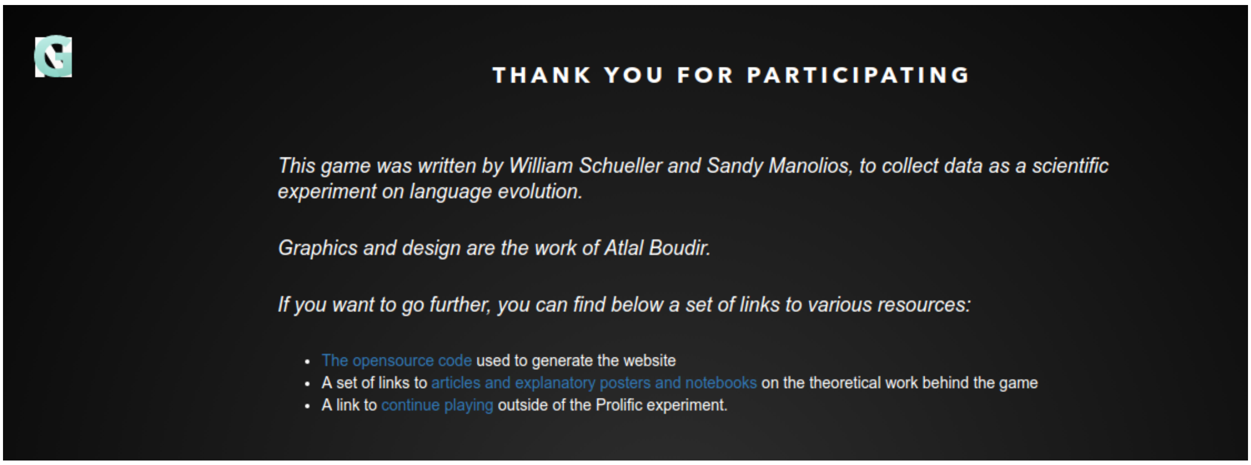
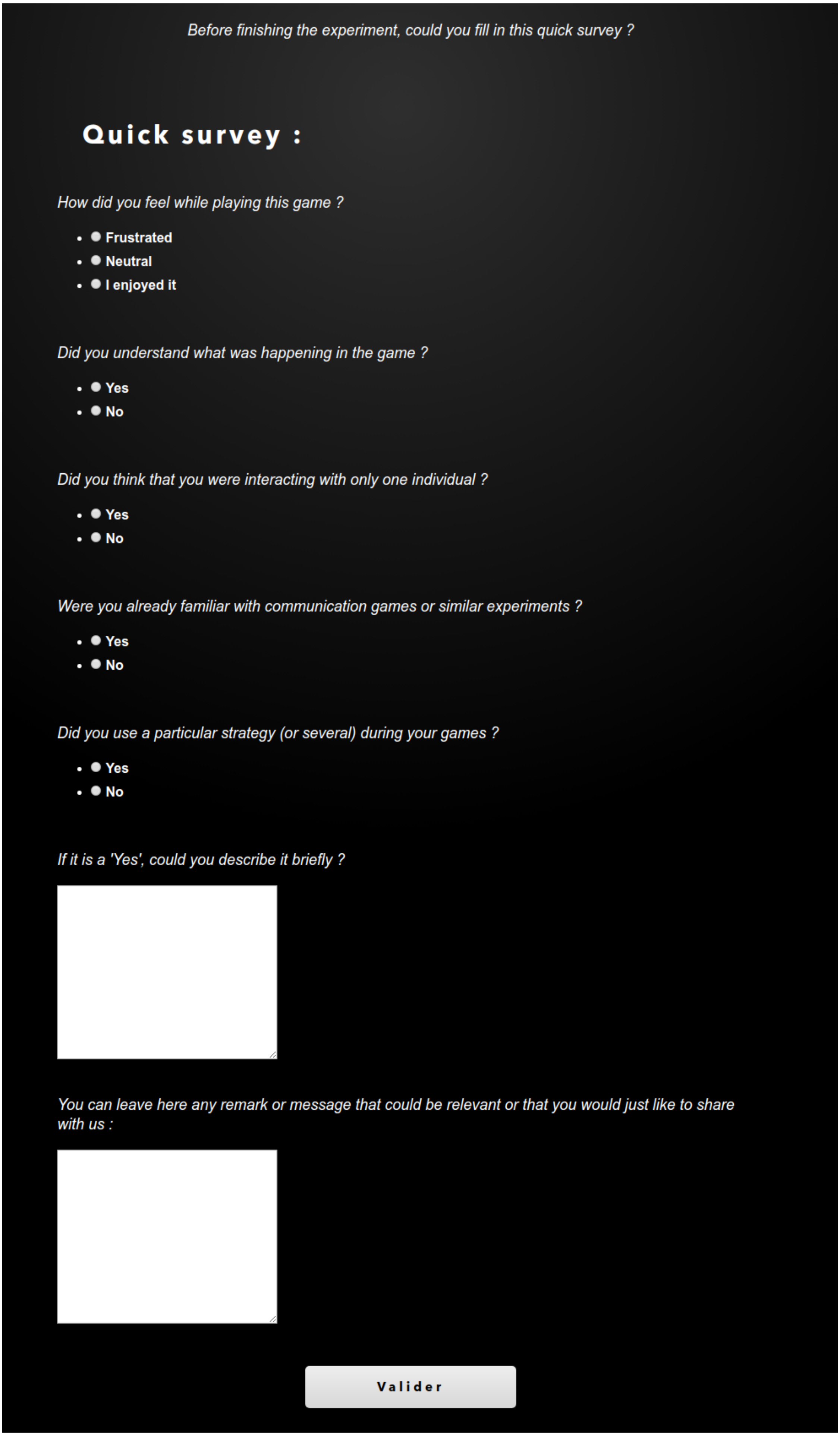


Figure B.20: Bonus screen



Before finishing the experiment, could you fill in this quick survey ?

Quick survey :

How did you feel while playing this game ?

- **Frustrated**
- **Neutral**
- **I enjoyed it**

Did you understand what was happening in the game ?

- **Yes**
- **No**

Did you think that you were interacting with only one individual ?

- **Yes**
- **No**

Were you already familiar with communication games or similar experiments ?

- **Yes**
- **No**

Did you use a particular strategy (or several) during your games ?

- **Yes**
- **No**

If it is a 'Yes', could you describe it briefly ?

You can leave here any remark or message that could be relevant or that you would just like to share with us :

Valider

Figure B.21: End survey

Bibliography

Aldous, David J. (1985). "Exchangeability and related topics". In: *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, pp. 1–198. DOI: 10.1007/bfb0099421.

Barabási, Albert-László and Réka Albert (1999). "Emergence of scaling in random networks". In: *science* 286.5439, pp. 509–512.

Baronchelli, Andrea (2006). "Statistical mechanics approach to language games". PhD thesis. PhD thesis," Sapienza" Univ. of Rome.

– (2016). "A gentle introduction to the minimal Naming Game". In: *Belgian Journal of Linguistics* 30.1, pp. 171–192. DOI: 10.1075/bjl.30.08bar.

Baronchelli, Andrea, Luca Dall'Asta, et al. (2005). "Strategies for fast convergence in semiotic dynamics". In: *arXiv preprint physics/0511201*.

Baronchelli, Andrea, Maddalena Felici, et al. (2006). "Sharp transition towards shared vocabularies in multi-agent systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2006.06, P06014–P06014. DOI: 10.1088/1742-5468/2006/06/p06014.

Baronchelli, Andrea, Tao Gong, et al. (2010). "Modeling the emergence of universality in color naming patterns". In: *Proceedings of the National Academy of Sciences* 107.6, pp. 2403–2407.

Barto, Andrew G, Satinder Singh, and Nuttapong Chentanez (2004). "Intrinsically motivated learning of hierarchical collections of skills". In: *Proceedings of the 3rd International Conference on Development and Learning*, pp. 112–19.

Berlyne, Daniel E (1960). *Conflict, arousal, and curiosity.* McGraw-Hill Book Company. DOI: 10.1037/11164-000.

– (1965). "Structure and direction in thinking." In:

Beuls, Katrien and Luc Steels (2013). "Agent-based models of strategies for the emergence and evolution of grammatical agreement". In: *PloS one* 8.3, e58960.

Bleys, Joris et al. (2009). "The grounded colour naming game". In: Institute of Electrical and Electronics Engineers.

Blom, Gunnar, Lars Holst, and Dennis Sandell (1994). *Problems and Snapshots from the World of Probability*. Springer New York. DOI: 10.1007/978-1-4612-4304-5.

Bloom, Paul (2002). *How children learn the meanings of words*. MIT press.

Blythe, Richard A., Andrew D. M. Smith, and Kenny Smith (2016). "Word learning under infinite uncertainty". In: *Cognition* 151, pp. 18–27. DOI: 10.1016/j.cognition.2016.02.017.

Blythe, Richard A., Kenny Smith, and Andrew D. M. Smith (2010). "Learning Times for Large Lexicons Through Cross-Situational Learning". In: *Cognitive Science* 34.4, pp. 620–642. DOI: 10.1111/j.1551-6709.2009.01089.x.

Bubeck, Sébastien, Nicolò Cesa-Bianchi, et al. (2012). "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1, pp. 1–122. ISSN: 1935-8237. DOI: 10.1561/2200000024.

Centola, Damon and Andrea Baronchelli (2015). "The spontaneous emergence of conventions: An experimental study of cultural evolution". In: *Proceedings of the National Academy of Sciences* 112.7, pp. 1989–1994.

Clement, Benjamin et al. (2015). "Multi-Armed Bandits for Intelligent Tutoring Systems." In: *Journal of Educational Data Mining* 7.2, pp. 20–48.

Cornudella Gaya, Miquel (2017). "Autotelic Principle: the role of intrinsic motivation in the emergence and development of artificial language". PhD thesis. Université de recherche Paris Sciences et Lettres.

Cornudella, Miquel and Thierry Poibeau (2015). "Language Emergence in a Population of Artificial Agents Equipped with the Autotelic Principle". In: *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*. Association for Computational Linguistics. DOI: 10.18653/v1/w15-2407.

Csikszentmihalyi, Mihaly (1991). *Flow: The Psychology of Optimal Experience*. New York, NY: Harper Perennial.

Cuskley, Christine (2018). "Alien symbols for alien language: iterated learning in a unique, novel signal space". In: *The Evolution of Language: Proceedings of the 12th International Conference (EVOLANGXII)*. Ed. by C. Cuskley et al. NCU Press. DOI: 10.12775/3991-1.018. URL: http://evolang.org/torun/proceedings/papertemplate.html?p=42.

Cuskley, Christine et al. (2017). "The regularity game: Investigating linguistic rule dynamics in a population of interacting agents". In: *Cognition* 159, pp. 25–32.

Dall'Asta, Luca et al. (2006). "Nonequilibrium dynamics of language games on complex networks". In: *Physical Review E* 74.3. DOI: 10.1103/physreve.74.036105.

De Boer, Bart (2001). *The origins of vowel systems*. Vol. 1. Oxford University Press on Demand.

De Vylder, Bart (2007). "The evolution of conventions in multi-agent systems". In: *Unpublished doctoral dissertation, Vrije Universiteit Brussel, Brussels*.

Erdos, Paul and Alfréd Rényi (1960). "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1, pp. 17–60.

Fagyal, Zsuzsanna et al. (2010). “Centers and peripheries: Network roles in language change”. In: *Lingua* 120.8, pp. 2061–2079. doi: 10.1016/j.lingua.2010.02.001.

Ferdinand, Vanessa and Matt Spike (2016). “Word Learners Regularize Synonyms And Homonyms Similarly”. In: *The Evolution of Language: Proceedings of the 11th International Conference (EVOLANGX11)*. Ed. by S. G. Roberts et al. Online at http://evolang.org/neworleans/papers/82.html.

Galantucci, Bruno and Simon Garrod (2012). *Experimental Semiotics: Studies on the emergence and evolution of human communication*. Vol. 45. John Benjamins Publishing.

Garrod, Simon et al. (2007). “Foundations of Representation: Where Might Graphical Symbol Systems Come From?” In: *Cognitive Science* 31.6, pp. 961–987. doi: 10.1080/03640210701703659.

Grieve, Jack, Andrea Nini, and Diansheng Guo (2018). “Mapping Lexical Innovation on American Social Media”. In: *Journal of English Linguistics*, p. 0075424218793191.

Hidaka, Shohei, Takuma Torii, and George Kachergis (2017). “Quantifying the impact of active choice in word learning”. In: Cognitive Science Society.

Howard, Ian S and Piers Messum (2011). “Modeling the development of pronunciation in infant speech acquisition”. In: *Motor Control* 15.1, pp. 85–117.

Hunt, JMcVLevine (1965). “Intrinsic motivation and its role in psychological development”. In: *Nebraska symposium on motivation*. Vol. 13. University of Nebraska Press, pp. 189–282.

Kachergis, George, Chen Yu, and Richard M Shiffrin (2013). “Actively learning object names across ambiguous situations”. In: *Topics in Cognitive Science* 5.1, pp. 200–213. doi: 10.1111/tops.12008.

Karjus, Andres et al. (2018). “Quantifying the dynamics of topical fluctuations in language”. In: *arXiv preprint arXiv:1806.00699*.

Kauffman, Stuart A (1996). “Investigations: The nature of autonomous agents and the worlds they mutually create”. In: Santa Fe Institute.

Kay, Paul et al. (2009). *The world color survey*. CSLI Publications Stanford.

Ke, Jinyun et al. (2002). “Self-organization and selection in the emergence of vocabulary”. In: *Complexity* 7.3, pp. 41–54.

Kirby, Simon (2001). “Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity”. In: *IEEE Transactions on Evolutionary Computation* 5.2, pp. 102–110.

Kirby, Simon, Hannah Cornish, and Kenny Smith (2008). “Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language”. In: *Proceedings of the National Academy of Sciences*.

Kirby, Simon, Tom Griffiths, and Kenny Smith (2014). “Iterated learning and the evolution of language”. In: *Current opinion in neurobiology* 28, pp. 108–114.

Kirby, Simon and James R. Hurford (2002). “The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model”. In: *Simulating the Evolution of Language*. Springer London, pp. 121–147. DOI: 10.1007/978-1-4471-0663-0_6.

Kuhl, Patricia K (2004). “Early language acquisition: cracking the speech code”. In: *Nature reviews neuroscience* 5.11, p. 831.

Lapeyre, Matthieu et al. (2014). “Poppy project: open-source fabrication of 3D printed humanoid robot for science, education and art”. In: *Digital Intelligence 2014*, p. 6.

Loreto, Vittorio, Andrea Baronchelli, et al. (2011). “Statistical physics of language dynamics”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2011.04, P04006. DOI: 10.1088/1742-5468/2011/04/p04006.

Loreto, Vittorio, Animesh Mukherjee, and Francesca Tria (2012). “On the origin of the hierarchy of color names”. In: *Proceedings of the National Academy of Sciences*.

Mocanu, Delia et al. (2013). “The twitter of babel: Mapping world languages through microblogging platforms”. In: *PloS one* 8.4, e61981.

Morin, Olivier, James Winters, Thomas F Müller, et al. (2018). “What smartphone apps may contribute to language evolution research”. In: *Journal of Language Evolution* 3.2, pp. 91–93. DOI: 10.1093/jole/lzy005.

Morin, Olivier, James Winters, Thomas Müller, et al. (2018). “The “Color Game” App”. In:

Moulin-Frier, Clément, Sao Mai Nguyen, and Pierre-Yves Oudeyer (2014). “Self-organization of early vocal development in infants and machines: the role of intrinsic motivation”. In: *Frontiers in psychology* 4, p. 1006.

Newman, Mark EJ (2005). “Power laws, Pareto distributions and Zipf’s law”. In: *Contemporary physics* 46.5, pp. 323–351. DOI: 10.1080/00107510500052444.

Oliphant, Michael (1999). “The Learning Barrier: Moving from Innate to Learned Systems of Communication”. In: *Adaptive Behavior* 7.3-4, pp. 371–383. DOI: 10.1177/105971239900700309.

Oliphant, Michael and John Batali (1997). “Learning and the emergence of coordinated communication”. In: *Center for research on language newsletter* 11.1, pp. 1–46.

Oller, D Kimbrough (2000). *The emergence of the speech capacity*. Psychology Press.

Oudeyer, Pierre-Yves (2006). *Self-organization in the evolution of speech*. Vol. 6. 6. OUP Oxford.

Oudeyer, Pierre-Yves and Frederic Delaunay (2008). “Developmental exploration in the cultural evolution of lexical conventions”. In: *8th international conference on epigenetic robotics: Modeling cognitive development in robotic systems*.

Oudeyer, Pierre-Yves and Frédéric Kaplan (2006). “Discovering communication”. In: *Connection Science* 18.2, pp. 189–206.

– (2009). “What is intrinsic motivation? A typology of computational approaches”. In: *Frontiers in neurorobotics* 1, p. 6.

Oudeyer, Pierre-Yves, Frédéric Kaplan, and Verena V. Hafner (2007). “Intrinsic Motivation Systems for Autonomous Mental Development”. In: *IEEE Transactions on Evolutionary Computation* 11.2, pp. 265–286. doi: 10.1109/tevc.2006.890271.

Oudeyer, Pierre-Yves and Linda B Smith (2016). “How evolution may work through curiosity-driven developmental process”. In: *Topics in Cognitive Science* 8.2, pp. 492–502.

Partridge, Eric et al. (2015). “Young children’s self-directed information gathering on touchscreens”. In: *Proceedings of the 37th Annual Conference of the Cognitive Science Society, Austin, TX. Cognitive Science Society*.

Pucci, Lorenzo, Pietro Gravino, and Vito D. P. Servedio (2014). “Modeling the Emergence of a New Language: Naming Game with Hybridization”. In: *Self-Organizing Systems*. Springer Berlin Heidelberg, pp. 78–89. doi: 10.1007/978-3-642-54140-7_7.

Puglisi, Andrea, Andrea Baronchelli, and Vittorio Loreto (2008). “Cultural route to the emergence of linguistic categories”. In: *Proceedings of the National Academy of Sciences* 105.23, pp. 7936–7940.

Raviv, Limor, Antje Meyer, and Shiri Lev-Ari (2019). “Compositional structure can emerge without generational transmission”. In: *Cognition* 182, pp. 151–164. doi: 10.1016/j.cognition.2018.09.010.

Ren, Wei, Randal W Beard, and Ella M Atkins (2005). “A survey of consensus problems in multi-agent coordination”. In: *American Control Conference, 2005. Proceedings of the 2005*. IEEE, pp. 1859–1864.

Ryan, Richard M. and Edward L. Deci (2000). “Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions”. In: *Contemporary Educational Psychology* 25.1, pp. 54–67. doi: 10.1006/ceps.1999.1020.

Schaekermann, Mike et al. (2018). “Resolvable vs. Irresolvable Disagreement”. In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW, pp. 1–19. doi: 10.1145/3274423.

Schmidhuber, J. (1991). “Curious model-building control systems”. In: *[Proceedings] IEEE International Joint Conference on Neural Networks*. IEEE. doi: 10.1109/ijcnn.1991.170605.

Schueller, William, Vittorio Loreto, and Pierre-Yves Oudeyer (2018). “Complexity Reduction in the Negotiation of New Lexical Conventions”. In: *40th Annual Conference of the Cognitive Science Society (CogSci 2018)*. Madison, WI, United States.

Schueller, William and Pierre-Yves Oudeyer (2015). “Active learning strategies and active control of complexity growth in naming games”. In: *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE. doi: 10.1109/devlrn.2015.7346144.

– (2016). “Active Control of Complexity Growth in Naming Games: Hearer’s Choice”. In: *EVOLANG 2016*. Proceedings for the 11th

International Conference on the Evolution of Language (EvoLang XI). New Orleans, United States.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961.

Smith, Kenny, Andrew DM Smith, and Richard A Blythe (2011). "Cross-situational learning: An experimental study of word-learning mechanisms". In: *Cognitive Science* 35.3, pp. 480–498.

Spranger, Michael (2012). "The co-evolution of basic spatial terms and categories". In: *Experiments in cultural language evolution* 3, p. 111.

Steels, Luc (1995). "A self-organizing spatial vocabulary". In: *Artificial life* 2.3, pp. 319–332. DOI: 10.1162/artl.1995.2.3.319.

– (2000). "Language as a complex adaptive system". In: *International Conference on Parallel Problem Solving from Nature*. Springer, pp. 17–26.

– (2004). "The autotelic principle". In: *Embodied artificial intelligence*. Springer, pp. 231–242.

– (2015). *The Talking Heads experiment: Origins of words and meanings*. Vol. 1. Language Science Press.

Steels, Luc and Frédéric Kaplan (1998). "Stochasticity as a source of innovation in language games". In: *Proceedings of the Conference on Artificial Life VI (Alife VI)(Los Angeles)*, pp. 368–76.

Steels, Luc and Pieter Wellens (2007). "Scaffolding language emergence using the autotelic principle". In: *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*. IEEE, pp. 325–332.

Steels, Luc, Tony Belpaeme, et al. (2005). "Coordinating perceptually grounded categories through language: A case study for colour". In: *Behavioral and brain sciences* 28.4, pp. 469–488.

Thrun, Sebastian (1995). "Exploration in active learning". In: *Handbook of Brain Science and Neural Networks*, pp. 381–384.

Tria, Francesca, Vittorio Loreto, and Vito Servedio (2018). "Zipf's, Heaps' and Taylor's Laws are Determined by the Expansion into the Adjacent Possible". In: *Entropy* 20.10, p. 752.

Tria, Francesca, Vito D. P. Servedio, et al. (2015). "Modeling the Emergence of Contact Languages". In: *PLOS ONE* 10.4. Ed. by Matjaz Perc, e0120771. DOI: 10.1371/journal.pone.0120771.

Van Trijp, Remi (2012). "The evolution of case systems for marking event structure". In: *Experiments in cultural language evolution* 3, pp. 169–205.

Verhoef, Tessa, Simon Kirby, and Bart De Boer (2014). "Emergence of combinatorial structure and economy through iterated learning with continuous acoustic signals". In: *Journal of Phonetics* 43, pp. 57–68.

Vogt, Paul and Hans Coumans (2003). "Investigating social interaction strategies for bootstrapping lexicon development". In: *Journal of Artificial Societies and Social Simulation* 6.1.

Vollmer, Anna-Lisa et al. (2014). "Studying the co-construction of interaction protocols in collaborative tasks with humans". In: *4th In-*

ternational Conference on Development and Learning and on Epigenetic Robotics. IEEE. doi: 10.1109/devlrn.2014.6982983.

Vredenburgh, Christopher and Tamar Kushnir (2015). "Young Children's Help-Seeking as Active Information Gathering". In: *Cognitive science* 40.3, pp. 697–722. doi: 10.1111/cogs.12245.

Vylder, Bart De and Karl Tuyls (2006). "How to reach linguistic consensus: A proof of convergence for the naming game". In: *Journal of Theoretical Biology* 242.4, pp. 818–831. doi: 10.1016/j.jtbi.2006.05.024.

Wellens, Pieter (2012). "Adaptive strategies in the emergence of lexical systems". In: *Unpublished doctoral dissertation, Vrije Universiteit Brussel, Brussels*.

West, G. B. (1997). "A General Model for the Origin of Allometric Scaling Laws in Biology". In: *Science* 276.5309, pp. 122–126. doi: 10.1126/science.276.5309.122.

Winckler, Bruno (2009). *Recueil de blagues mathématiques et autres curiosités*. Éditions Ellipses. isbn: 9782729864514.

Wittgenstein, Ludwig (1953). *Philosophische Untersuchungen I Philosophical investigations (GEM. Anscombe & R. Reesh, Eds.,)*

Zipf, George Kingsley (1949). *Human behavior and the principle of least effort: An introduction to human ecology*. Cambridge, MA.

Table of symbols

$\mathfrak{M}$	Meaning space
M	Size of the meaning space
$\mathfrak{M}_k$	Set of known meanings
μ	Number of known meanings
$\mathfrak{M}_u$	Set of unknown meanings
m_i	Element of the meaning space
$\mathfrak{W}$	Word space
W	Size of the word space
w_i	Element of the word space
m_S	Meaning picked by the speaker, or topic
m_H	Meaning interpreted by the hearer
S	Speaker (one of the two possible roles in a Naming Game interaction)
H	Hearer (one of the two possible roles in a Naming Game interaction)
t	time, in total number of interactions
$S(t)$	Theoretical Communicative Success measure
t_{conv}	Time when reaching a global agreement: every agent has the same completed lexicon.
$S_{LAPS}(t)$	Local Approximated Probability of Success measure
$S_O(t)$	Observed Communicative Success
$N_l(t)$	Local Complexity measure
N_l^{max}	Maximum of Local Complexity measure
t_{max}	Time when reaching the maximum of Local Complexity measure
$N_d(t)$	Global Complexity measure
N_d^{max}	Maximum of Global Complexity measure
t_{GC}	Time when reaching the maximum of Local Complexity measure
N_{inv}	Number of inventions during a Naming Game
n_{ch}	Number of chunks for the Chunks strategy, depending on N and W
α_{ST}	Parameter for the Success Threshold, normalized (divided by N)
n_{MC}	Parameter for the minimal counts strategy
$\widetilde{n}_{MC}$	Parameter for the Minimal Counts strategy, normalized (divided by N)
τ	Time scale parameter for LAPSmax and Coherence strategies

Table of Abbreviations and Acronyms

MAB	Multi-Armed Bandit
LAPS	Local Approximated Probability of Success
ATC	Active Topic Choice
RTC	Random Topic Choice
AP	Acceptance Policy
NG	Naming Game
BLIS	Basic Lateral Inhibition Stategy, a vocabulary update policy
TCS	Theoretical Communicative Success measure
LC	Local Complexity measure
GC	Global Complexity measure