THSim v3.2: The Talking Heads simulation tool

Paul Vogt

Induction of Linguistic Knowledge, Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands.
Language Evolution and Computation Research Unit
University of Edinburgh, UK.
`paulv@ling.ed.ac.uk`

Abstract. The field of language evolution and computation may benefit from using efficient and robust simulation tools that are based on widely exploited principles within the field. The tool presented in this paper is one that could fulfil such needs. The paper presents an overview of the tool – THSim v3.2 – and discusses some research questions that can be investigated with it.

1 Introduction

The field of language evolution and computation has become increasingly popular in the alife community, see [4, 10] for recent overviews. Up to now, research has focused on the origins and evolution of simple symbolic communication systems (or *lexicons*), e.g., [3, 7, 11, 13, 14, 16, 18] and on the evolution of syntax, e.g., [1, 2, 9]. The bulk of the research so far have assumed a predefined meaning space [1, 2, 9, 7, 11, 14, 18]; an assumption that inevitably leads to the *symbol grounding problem* [6], which relates to the question how agents interpret symbols meaningfully. Relatively few experiments are known that tackles the symbol grounding problem in relation to lexicon formation, both in simulations [3, 12, 16] and robotic platforms [14, 15]. The simulation tool *THSim v3.2* presented in this paper has been designed to study various aspects of grounded lexicon formation.

One of the most familiar experiments on lexicon grounding is the *Talking Heads* experiment [14]. In this experiment a population of agents developed a lexicon from scratch by engaging in language games. The agents were embodied as pan-tilt cameras looking at a white-board on which geometrical figures were pasted. The experiment was set up such that human users could interact with the experiment through the Internet by launching agents, controlling their whereabouts and altering their lexical entries. Although this was an interesting feature that showed how the system could deal with the resultant open dynamics, its disadvantage was that the experiments were uncontrolled and therefore few scientifically valid experiments could be done. Nevertheless, some interesting controlled experiments were done with the Talking Heads, see, e.g., [8, 14]. However, many open questions remain that could be studied with the Talking Heads.

To study these questions in a cheap, efficient and open manner, a new Talking Heads simulation tool has been designed. The development of this tool was part of a Dutch project to develop on-line education programmes in Knowledge Technology and has been made accessible for students and teachers associated with the project.[1] Currently, version 3.2 is available on the web for everyone who wishes to use it.[2]

This paper presents the main the functionalities of the tool. More details are available in the tool's manual. The next section presents a detailed overview of the tool's functions. Section 3 presents some open questions that can be studied using THSim.

2 The Talking Heads simulation tool

The Talking Heads simulation tool (THSim) has been designed as a tool to investigate various aspects of language evolution in a controlled and robust manner. In addition, it is a tool that allows visualisation of the ongoing processes, which is useful for a researcher using the tool or for users who are interested in learning about simulating language evolution, such as (under)graduate students. Although written in Java, which makes the simulation platform independent, the tool is sufficiently fast. The software is designed in an object-oriented fashion to make changes in the program relatively straightforward. In this section, an overview of the tool is presented.

2.1 The user interface

When THSim is started with the user-interface, a canvas is displayed on the computer screen with four different windows, see Figure 1. Clock-wise from the top-left these windows are called 'GEOM world', 'Control', 'Statistics' and 'Language games'. The 'GEOM world' [14] is the environment of the population. This environment contains 10 different geometrical figures (such as rectangles, circles, triangles and various regular and irregular polygons) of randomly generated, but distinctive colours that can be displayed on the screen. When a language game is played, a given number of objects are generated randomly and displayed on the screen. These objects constitute the context of the language game. Features relating to these objects are handed to the agents that play the language game. These features are generated such that the information they contain could be extracted from a camera image in a similar way as in the original Talking Heads experiment.

The 'Control' panel allows the user to set various parameters and to control the simulation. Some of these parameters will be discussed briefly in Section 3.

---

[1] The project is called LOK (*national education web knowledge technology*), and its web-pages – which are only available in Dutch – can be found on http://www.ou.nl/lok/.

[2] http://www.ling.ed.ac.uk/~paulv/thsim.html

nGames 1000 nAgents 2 cxtSize 4
pNoise 0.0 pWC 1.0 etaN 0.9 etaS 0.9
pAdultS 1.0 pAdultH 0.0 ILM ☐ nIter 1
S2S ☑ incrPop ☐ growth 2 maxAgents 20
varGames ☐ pOG 0.0 pGG 1.0 adaptSG ☑
uPType centre-of-mass
Select features: R ☑ G ☑ B ☑ A ☑ X ☑ Y ☑
gameType guessing game
foa 4 fCol ☑ uScore ☑ log ☐ lex ☐

Iteration 0, language game 1000, type=g
Speaker A3 topic=1
Meaning: M7
Hearer A2 topic=1
Meaning: M34
Utterance = 'mofeji', language game succeeds.

Fig. 1. A screen shot of THSim with default parameter settings shown in the control panel.

The 'Statistics' window shows a graph displaying some relevant statistics of the simulation that is being played. The statistics that are shown relate to the communicative success and the discriminative success of the simulation. Communicative success measures the effectiveness of inter-agent communication, while discriminative success measures the effectiveness of the conceptualisation or meaning formation.

The 'Language games' window displays some information about each language game that is being played. This information include the agents that participate, the features relating to the topic of the language game (open boxes), the categories that the agents use to categorise the topic (coloured boxes), the way the agents name the topic and the outcome of the language game.

2.2 Basic functionality: language games

THSim's functionality is based around the language game model (see, e.g., [13, 14]). A language game is played by two agents who are selected from the population. Both agents jointly attend to a scene that is generate by the GEOM world and which constitutes the context of the game. The speaker of the language game selects one object from the context as the topic and tries to produce an utterance to name this topic. Before producing an utterance, however, the speaker has to categorise the object. This is done using a variant of the *discrimination game* (see, e.g., [14, 15]). After having produced an utterance, the hearer of the game tries to interpret the utterance. It first has to play one or more discrimination games to construct a meaning on which it can match the utterance. After the hearer interprets the utterance, both agents adapt their lexicons depending on certain constraints. One constraint is the type of language game that the agents play. Currently three types of language games have been implemented in THSim: the observational game, the guessing game and the selfish game. Below follows a detailed description of these three games; for more details consult [16, 18]. But before explaining the language games, the discrimination game is explained.

Discrimination game The objective of the discrimination game is for each individual agent to find a categorisation of the topic that distinguishes this topic from all other objects in the context. By playing a number of discrimination games, each agent a gradually constructs its ontology $\mathcal{O}_a$. An ontology is a set of categories $\mathcal{O}_a = \{c_0, \ldots, c_p\}$. The categories c_i are represented as prototypes $\mathbf{c}_i$, which are points in an n-dimensional meaning space. At the start of an agent's lifetime, the ontology is empty. The game basically consists of four steps: feature extraction, categorisation, discrimination and adaptation.

Feature extraction: When the agents attend to the scenery of the GEOM world, they detect a number of features $f_k \in [0, 1]$ for each object in the context. Hence, each object can be described by an n-dimensional feature vector $\mathbf{f}_i = (f_1, \ldots, f_n)$, where n equals the number of features an agent extracts and which is equal to the dimension of the meaning space. Currently, six features are implemented: the Red, Green and Blue components of the RGB colour space, the position on the X-axis, on the Y-axis and the shape feature (A). Shape feature A is calculated by the function $2 \cdot (\frac{A_o}{A_{bb}} - \frac{1}{2})$, where $\frac{A_o}{A_{bb}}$ is the filling-ratio of the object's area A_o and the area of the object's bounding-box A_{bb} – which is the smallest rectangle that can be drawn around the object. This feature produces a value that indicates the shape of the object; e.g., all rectangles have values of 1.0, circles have values of 0.57 and triangles have values of 0.0.

Categorisation: In the categorisation phase, a category $c_j \in \mathcal{O}_a$ is searched for each feature vector $\mathbf{f}_i$ such that the Euclidean distance $||\mathbf{f}_i - \mathbf{c}_j||$ is smallest. This is the *1-nearest neighbourhood search* [5].

Discrimination: In this phase it is verified whether the category for the topic is distinctive from the categories relating to the other objects in the context. If no such category exists, the discrimination game fails and the ontology has to be expanded (see Adaptation). Otherwise, the discrimination game is a success and the resulting *distinctive category*, c_d, is forwarded to the production or interpretation phase of the language game.

Adaptation: At the end of the discrimination game, the ontology of the agent is adapted according to the outcome of the game. There are two possibilities:

1. **Failure:** If the game is a failure, the ontology is expanded with a new category for which the feature vector $\mathbf{f}_t$ of the topic is used as an exemplar.
2. **Success:** In case of success, the distinctive category c_d is adjusted to make it a more representative sample of the objects it categorised. There are currently three implementations of this adaptation and the default calculates the centre-of-mass of the feature vectors it represents and is defined as follows:

$$\mathbf{c}_d = \frac{U(c_d) \cdot \mathbf{c}_d + \mathbf{f}_t}{U(c_d) + 1} \tag{1}$$

In this equation $U(c_d)$ is the frequency with which category c_d was used as a distinctive category.

Three different language games In order to investigate the impact of non-verbal social interactions on lexicon formation, three different language games have been designed: the observational game (OG), guessing game (GG) and selfish game (SG).

In a language game, a speaker agent tries to verbalise the meaning of the topic, which a hearer agent tries to interpret. Depending on their success, both agents adapt their lexicons. New lexical elements may be constructed by invention or adoption, and association scores indicating the effectiveness of elements are increased or decreased. The way the agents evaluate the effectiveness of the game and the way scores are adapted depend on the type of language game they play.

OG: The speaker informs the hearer which topic it selected before the verbal interaction. Scores are adapted following Hebbian learning.

GG: The speaker does not inform the hearer about the topic prior to the verbal communication, but verifies if the hearer guessed the right topic. The scores are then adapted similar to reinforcement learning.

SG: The hearer is not informed about the topic, nor is the success of the game evaluated. Both agents adapt their scores in a way that relates to Bayesian learning.

Each agent $a \in \mathcal{A}$ has a private lexicon $\mathcal{L}_a = \{l_0, \ldots, l_q\}$. A lexical element l_i is defined as a triplet containing a word w_i, a meaning m_i and an association score σ_i, i.e. $l_i = \langle w_i, m_i, \sigma_i \rangle$. A word is constructed from 1 to 3 consonant-vowel pairs, where the consonants and vowels are taken from a given alphabet.[3] The meanings are categories that were used in a language game at least once, which means they must have been used distinctively at least once. The association score is a real value between 0 and 1, indicating the effectiveness of the lexical element. There are two different implementations of the association scores: one is

---

[3] The words that are constructed can look like "pi", "wilo" and "wateve" for example.

St.	Game	Description
1.	all	Two agents are randomly selected from the population $\mathcal{A}$, one becomes the speaker S, the other becomes hearer H.
2.	all	The agents 'observe' the GEOM world where a context $C = \{o_0, \ldots, o_r\}$ of geometrical figures o_i is constructed. In addition, a focus of attention $F \subseteq C$ is established by selecting a number of objects randomly from the context. F and C are shared by both agents.
3.	all	S selects an $o_i \in F$ as the topic t_S of the language game.
4.	**OG**	S informs H what object it selected as topic.
5.	all	S plays a discrimination game to find a distinctive category c_d for the topic. If the discrimination game fails, the language game fails too and stops here. Note that discrimination games are always played relative to the context C.
6.	all	S produces an utterance $u = w_i$, where the word w_i is from a lexical element $l_i \in \mathcal{L}_S$ for which $m_i = c_d$ and $\sigma_i \geq \sigma_j$ for all other elements $l_j \in \mathcal{L}_S$ for which $m_j = c_d$. If no such element is found, a new element $l = \langle w, c_d, 0.01 \rangle$ is added to the lexicon $\mathcal{L}_S$, where w is a newly invented word. This construction occurs only with a certain *word-creation probability* pWC. If no utterance is produced, the language game fails and stops here.
7.	all	When, and if, H receives the utterance u, it plays one or more discrimination games. If it does not know the topic, as is the case in the GG and SG, it does so for every $o_i \in F$. The results of the discrimination game(s) is stored in the distinctive category set D. If $D = \emptyset$, the language game fails and stops here.
8.	all	If $D \neq \emptyset$, H tries to interpret the utterance by searching an element $l_i \in \mathcal{L}_H$ for which $w_i = u$, $m_i \in D$ and $\sigma_i \geq \sigma_j$ for other elements $l_j \in \mathcal{L}_H$ with $w_j = u$ and $m_j \in D$. The object that was categorised with m_i becomes H's topic t_H. (If H was informed about the topic as in point 4, H can only find one matching element and the above still holds.)
9.	**GG**	If H guessed a topic, it informs S which object it guessed. S verifies whether this is the same topic and provides H with *corrective feedback*. If $t_S = t_H$, the GG was successful. If H's topic was different from S's, there was a mismatch in reference and S presents H the topic.
10.		At this point the language game finishes and both agents adapt their lexicons. The adaptations differ for the three different games:
	OG	If H found a lexical element to cover u, the OG was successful and both agents increase the association score of the used element l_i by $\sigma_i = \eta \cdot \sigma_i + 1 - \eta$, where $\eta \in \langle 0, 1 \rangle$ is a learning parameter (at default $\eta = 0.9$). In addition, the association scores of competing elements l_j are laterally inhibited by $\sigma_j = \eta \cdot \sigma_j$. An element l_j is competing if $(w_j = u) \wedge (m_j \neq m_i)$ or if $(w_j \neq u) \wedge (m_j = m_i)$. The OG fails if H does not know the word in relation to the distinctive category of the topic. In that case, H adopts u and adds the element $l = \langle u, c_d, 0.01 \rangle$ to $\mathcal{L}_H$. In turn, S lowers the association score of the used element l_i by $\sigma_i = \eta \cdot \sigma_i$.
	GG	The adaptation for the GG is basically the same as in the OG: In case of success the score of the used element is increased while competing elements are inhibited. In case of failure, S lowers the association score and, if necessary, H adopts u and associates it with the topic, which has then

St.	Game	Description
10.	**GG**	been indicated by S. Note that the GG fails if $t_S \neq t_H$, which is the case when H does not know u or when it misinterprets u. In case H misinterprets u, it also decreases the association score of the used element.
	SG	In the SG no success or failure is evaluated. S increases and the association score of the used element as in the case of a successful observational game and laterally inhibits competing elements. H increases the association scores for elements that are in the focus of attention F (i.e. elements l_i for which $w_i = u$ and $m_i \in D$). If there is a meaning $m \in D$ that is not yet lexicalised, the association $l = \rangle u, m, 0.01\langle$ is added to $\mathcal{L}_H$ first. Competing elements that fall outside the scope of F are laterally inhibited as before.

Table 1. A score-base description of the three language games. The first column numbers the steps taken, the second column indicates for which type of game the steps are applied, and the final column describes the step.

referred to as *score-based*, the other as *usage-based*. The difference will be made clear shortly. At the start of an agent's lifetime $\mathcal{L}_a = \emptyset$.

Given the discrimination game and the lexicon, the *score-based* language games are implemented as outlined in Table 1.

In the *usage-based* language games the association scores σ_i are calculated by $\sigma_i = \frac{U(w \wedge m_i)}{U(w)}$, where $U(w \wedge m_i)$ is the usage frequency of the co-occurrence of word w and meaning m_i. The denominator of this equation $U(w)$ is the usage frequency of word w disregarding which meaning it was used. The demoninator may be omitted by the hearer H as it only tries to interpret one utterance. The speaker S, however, considers different words when trying to produce an utterance.

The way the usage frequencies $U(w \wedge m_i)$ are updated at the end of a language game differs for the different games. For the OG and GG, $U(w \wedge m_i)$ is increased by 1 only if the association was used in a successful game, while $U(w)$ is increased by 1 everytime the word w is used, disregarding the game's success. For the SG, the update is more complicated: S increases both $U(w \wedge m_i)$ and $U(w)$ by 1 for the produced utterance u, and H increases $U(w \wedge m_i)$ and $U(w)$ by 1 for every element with $w = u$ and $m_i \in D$. As in the score-based SG, the association between w and $m_i \in D$ is added to the lexicon first it it does not yet exists, where initially $U(w \wedge m_i) = 0$. (Note that for the SG, the equation for calculating σ_i can be reformulated in terms of the conditional probability $P(m|w)$ that, given the occurrence of w, meaning m occurs, see [18].)

Iterated learning model One interesting aspect of studying language evolution is investigating how the language of one generation may be transmitted to the next generation. To study this aspect, the Iterated Learning Model (ILM) has been proposed [2, 9]. A population in the ILM contains a group of N adult agents and a group of N learner agents. The adults have passed the stage of learners and are supposed to have mastered the language. The learners learn from the adults by interacting with them using language games. The ILM has been adapted for THSim and iterates the following two steps:

1. K language games are played.
2. The adults are removed from the population and replaced by the learners. N new agents are placed in the learners group.

As a default setting, the speaker of a language game is always selected from the adult population and the hearer from the learner group, except in the first iteration where each agent is equally likely to be selected as speaker or hearer. It is possible to vary this setting as in the simulations reported in another paper in this volume [16].

3 Research questions

In this section, a few research questions will be discussed that can be investigated using THSim v3.2 as it is. The discussions will pose a research question, discuss briefly why the question is interesting and indicate what parameters should be set in THSim, apart from the default settings as shown in Fig. 1. When using THSim to investigate computational expensive settings, it is wise to start THSim without the user interface.

What is the influence of perceptual noise on lexicon formation?
In the default setting, the agents detect the features of the objects in GEOM world without noise and both agents in one game thus detect the same features. In a real world setting, such as the real Talking Heads, the perception of the world includes noise caused by physical factors such as varying lighting conditions, changes in temperature and the different locations of the agents participating in the game.

To investigate the effect of noise on the lexicon formation, it is possible to vary *pNoise* between 0 and 1. When this is done, each agent distorts the originally generated features f_i by $f_i' = f_i + \text{pNoise} \cdot (0.5 - X)$, where $X \in [0, 1]$ is a randomly generated real value.

Up to which population size can the simulations be scaled with or without using an incremental population growth?
As language societies are typically large and most simulations reported so far only have relatively low populations, it is interesting to investigate to what extent a population can increase without too much loss of performance.

It is possible to investigate this by changing the parameter *nAgents*, which effects the population size from the beginning of the simulation. Another realistic scenario would be to have language spread incrementally over a society. This can be investigated by setting the parameter *incrPop=true* in combination with *ILM=true* and *nIter>1*. The population then will grow incrementally in each new iteration by the parameter *growth*, until the maximum population size *maxAgents* is reached.

How should prototypes move through the meaning space?

One property of the discrimination game is that when it is played successfully, the prototype of the distinctive category is moved. The default implementation calculates the centre-of-mass of the feature vectors for prototypes which have been used distinctively, see Eq. 1. It is interesting to investigate the effect of varying the methods with which the prototypes are moved.

THSim allows the experimenter to alter the update type by varying the parameter *uPType*. The default is 'centre-of-mass', others are 'simulated annealing', 'walk' and 'none'. In simulated annealing, the prototype $\mathbf{c}$ is moved toward the feature vector $\mathbf{f}$ by $\Delta\mathbf{c} = (\mathbf{f} - \mathbf{c}) \cdot e^{-\frac{|\mathbf{f}-\mathbf{c}|}{T}}$, where $T = 1.0$ is the initial temperature which decays with $T = 0.9 \cdot T$ after each update of $\mathbf{c}$. In walk the prototype is moved by $\Delta\mathbf{c} = \epsilon \cdot (\mathbf{f} - \mathbf{c})$, where $\epsilon = 0.01$ is a constant step-size. In future releases of THSim, it will be possible to vary the constants T and ϵ.

What is the effect of varying game types?

As in [18], it is possible to compare the effect of applying the different language game types, while keeping the type fixed during one simulation. Additionally, it is possible to investigate the effect of varying the game type during one simulation. This is interesting as it is likely that human language users use various strategies to learn the meaning of words.

If one wishes to play either an OG, GG or SG during a simulation, one can select the game using the parameter *gameType*. If one wishes to investigate the effect of allowing agents to vary the games between OG, GG and SG in one simulation, one must set the parameter *varGames=true*. Then one can vary the probabilities with which an OG or GG is played by setting pOG and pGG. The probability with which the SG is played is automatically calculated by $pSG = 1 - pOG - pGG$. In order to investigate the added effect of learning in the SG with respect to the two other games, one can turn off the hearer's adaptation in the SG by deselecting *adaptSG*.

4 Conclusion

This paper presents an overview of the recently released simulation toolkit THSim. This toolkit, which is a simulation of the Talking Heads experiment, can be used by investigators who are interested in grounded lexicon formation. It must be stressed that every simulation starts with a population of agents that have no linguistic knowledge whatsoever, including knowledge of how to categorise their world. All this knowledge is bootstrapped during a simulation. Future releases will add other interesting functionalities, which include the formation of grammatical structures that is currently under construction [17].

Acknowledgements

This paper was written during a Visiting Research Fellowship at the Language Evolution and Computation Research Unit of the University of Edinburgh, awarded by the Royal Society of Edinburgh and the Caledonian Science Foundation. The members of LEC are thanked for their valuable comments and suggestions on this work.

References

1. J. Batali. Computational simulations of the emergence of grammar. In J. R. Hurford, M. Studdert-Kennedy, and C. Knight, editors, *Approaches to the Evolution of Language*, Cambridge, UK, 1998. Cambridge University Press.
2. H. Brighton. Compositional syntax from cultural transmission. *Artificial Life*, 8(1):25–54, 2002.
3. A. Cangelosi and D. Parisi. The emergence of "language" in an evolving population of neural networks. *Connection Science*, 10:83–93, 1998.
4. A. Cangelosi and D. Parisi, editors. *Simulating the Evolution of Language.* Springer, London, 2002.
5. T.M. Cover and P.E. Hart. Nearest neighbour pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27, 1967.
6. S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
7. J. R. Hurford. Biological evolution of the saussurean sign as a component of the language acquisition device. *Lingua*, 77,2:187–222, 1989.
8. F. Kaplan. *L'émergence d'un lexique dans une population d'agent autonomes.* PhD thesis, Laboratoire d'infromatique de Paris 6, 2000.
9. S. Kirby. Spontaneous evolution of linguistic structure: an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110, 2001.
10. S. Kirby. Natural language from artificial life. *Artificial Life*, 8(3), 2002.
11. M. Oliphant. The learning barrier: Moving from innate to learned systems of communication. *Adaptive Behavior*, 7 (3-4):371–384, 1999.
12. A. D. M. Smith. Establishing communication systems without explicit meaning transmission. In J. Kelemen and P. Sosík, editors, *Proceedings of the 6th European Conference on Artificial Life, ECAL 2001*, LNAI 2159, pages 381–390, Berlin Heidelberg, 2001. Springer-Verlag.
13. L. Steels. Emergent adaptive lexicons. In P. Maes, editor, *From Animals to Animats 4: Proceedings of the Fourth International Conference On Simulating Adaptive Behavior*, Cambridge Ma., 1996. The MIT Press.
14. L. Steels. *The Talking Heads Experiments. Volume 1. Words and Meanings.* Special pre-edition for LABORATORIUM, Antwerpen, 1999.
15. P. Vogt. Bootstrapping grounded symbols by minimal autonomous robots. *Evolution of Communication*, 4(1):89–118, 2000.
16. P. Vogt. Grounded lexicon formation without explicit meaning transfer: who's talking to who? In *Proceedings of ECAL*, 2003. this volume.
17. P. Vogt. Iterated learning and grounding: from holistic to compositional languages. In *Proceedings of the European Summer School in Logic, Language and Information*, 2003.
18. P. Vogt and H. Coumans. Investigating social interaction strategies for bootstrapping lexicon development. *Journal for Artificial Societies and Social Simulation*, 6(1), 2003. http://jasss.soc.surrey.ac.uk.