CS784 Term Project Presentation

Emergent Adaptive Lexicon

Nilesh Mishra
Rishabh Halakhandi
Dept. of CSE IITK

Overview

- Language as a Autonomous Evolving Adaptive system
- Luc Steels' experiment with software agents
- Details of the experiments
- Simulation results
- Conclusion
- References

Problem Statement

- Language can be seen as an emergent phenomenon.
- Language is a mass phenomenon actualized by different agents interacting with each other with no participant having a central view or control.
- It could spontaneously form itself once the underlying conditions are satisfied and could autonomously become more complex based on evolution, co-evolution, self-organization and level formation.

- In AI traditionally the designer carefully designs the rules and provides the ontology.
- Luc Steels’ work primarily tries to figure a mechanism through which the language evolves on its own and becomes more complex with little outside intervention (self organizes itself).
- In the learning by examples approach the necessary conceptualization is done by the teacher and the learning system is given positive and negative examples of the concept to be learned.

- It can be argued that the real intelligence in such systems lies in the teacher than the learning system.
- Also in many scenarios it is not always possible to get examples.
- In Steels' work the necessary conceptualization is done by the agents themselves when they develop new distinctions to discriminate objects in environment.
- The coherence in the system is reached by language interactions.

- The language emerging from these series of experiments has a number of human language like features, such as ambiguity, polysemy and synonymy.
- The system has positive reinforcement built into it. As the usage of a word increases the chances of success also increases.

The Actors

- The system primarily consists of three entities
  - The environment
  - A group of agents
  - A commentator
- The agents play a series of language games amongst themselves.

The Experiment*

- The experiment tests the hypothesis that language is an autonomous evolving adaptive system maintained by a group of distributed agents without central control.
- The system consists of an agent population (say 40 agents).
- In these 40 agents, at any given time only a subset of agents are active. The rest are sleeping. (say 20.)
- At the start of the game the agents are clear slates i.e. do not have any inbuilt ontology.

- The active agent subset plays language games which creates individual lexicons inside each agent.
- With a very small probability an active agent may go to sleep or an inactive agent may come alive. The lexicon of the agent coming out of sleep is the lexicon with which it went to sleep. (This makes the system an open system).

- The commentator keeps track of the system. It has access to the internal states of all the agents but no agent can query it.
- The environment acts as the mediator between the agents. It also selects the different groups (active, sleeping, context, speaker, topic and listener).

Jargon Buster*

- Feature: Concept created by the agents to distinguish/discriminate between objects and other agents present in the environment
- Discrimination tree: A graph which divides the continuous sensor values into subdomains leading to creation of features. The path taken gives the attribute and the node gives the value to the feature.
- Distinguishing feature set: A minimal set of features which will discriminate a topic from similar objects in the context.
- Word: A combination of alphabets.

- Utterance: A concatenation of words, where the word order is not important. It is the entity used for communication between agents
- Synonymy: More than one word mapped to a meaning (here attribute value pairs) in the agents lexicon.
- Ambiguity: Different agents associating different meanings to a word.
- Polysemy: More than one meanings attached to a word which can be narrowed down using the context.

Agent description

- An agent has an array of sensors attached with it.
- Each agent has a value for different sensor readings, which is initialized randomly at the start of the simulation.
- In the original experiment the number of sensors are limited and an agent associates a discrimination tree for each sensor to divide the sensor domain into non-overlapping, features.

- Each feature has an attribute and a set of feature values (range) associated with it.
- Instead of dividing the sensor domain we have increased the number of sensors. (original = 5, ours = 10).
- Every agent has a lexicon associated with it which is a mapping of feature sets and words.
- Each agent has a word generation module, a distinct feature discovery module and update modules for language game bookkeeping.

Distinct feature set*

- A distinct feature is a minimal set of features which uniquely identifies the topic from the context.
- The distinct feature set is found in an iterative manner where we generate different feature sets from the feature set of the topic in a combinatorial fashion.
- We stop at the level at which we get a distinct feature. E.g. If the distinct feature is a single feature then we calculate all the single feature distinct feature sets and do not go for distinct feature sets consisting of two or more features.

Selection criterion for feature

- If there exists more then one feature sets in the distinct feature set then we need to choose only one of them for further use in the language games.
- The selection criterion is in the following order:
  - To have the least number of features used, the smallest set is preferred.
  - In case of equal size the set in which the feature imply the smallest number of segmentation is preferred. This ensures that the most abstract features are chosen.
  - For equal depth of segmentation, we use the set for which the features have been used the most. This ensures that we develop a minimal set.

Feature Creation Process

- Due to the grounding experiments where the author uses physical robotic agents for simulation the number of sensors is limited.
- The goal of this mechanism is to provide an agent an adequate repertoire of features to discriminate between different agents in background.
- An elaborate mechanism is thus provided to generate enough features to distinguish the agent population.
- The mechanism starts with the association of a discrimination tree with each of the sensors.

- The discrimination tree divides the continuous domain into subdomains mapping the sensory inputs to discrete categories represented as features.
- Whenever the agent fails to get a distinct feature set, a new feature is created with a small probability.
- The new feature is created by randomly selecting a sensor channel and extending the discrimination tree associated with it by subdividing a distinction further.

The Lexicon

- The lexicon is a mapping between the different attribute-value pairs associated with an agents feature space (i.e. features generated by discrimination trees of different sensors) and words.
- Each lexicon is agent specific.
- The lexicon has a bidirectional association from feature space to words and vice versa.

- For each word entry there are pointers to different features associated with that word.
- The entries also contain the number of times that feature has been used as well as the number of time the feature’s use has resulted in success. These values are used for giving preference to different words in case of a conflict as well as to introduce a positive feedback loop into the system which leads to self-organization of the lexicon.

The Cover and Uncover function

- Cover function
  - It is used to generate utterances from feature sets.
  - Uses the lexicon mappings from feature sets to words and the use and success values to come up with an utterance for the input features set.
  - Used by the speaker in the language game

- Uncover function
  - It is used to get the possible feature set from the input utterance.
  - Uses the lexicon mappings from words to feature sets to get the possible feature set
  - Used by listener in a language game to come up with the possible feature set.

Word Generation/Addition

- When a language game fails because the speaker does not have a word for the corresponding feature it can generate a new word with a low probability.
- The new words are a pair of consonant and vowel chosen randomly and the relation added to the lexicon.
- When a language game fails because of hearer lacking adequate word feature mappings in its lexicon the new word obtained from speaker in the game is added to the lexicon with a relation with all the features in the distinct feature set calculated by hearer.

- Word addition creates ambiguity which are later resolved in further use in different context.

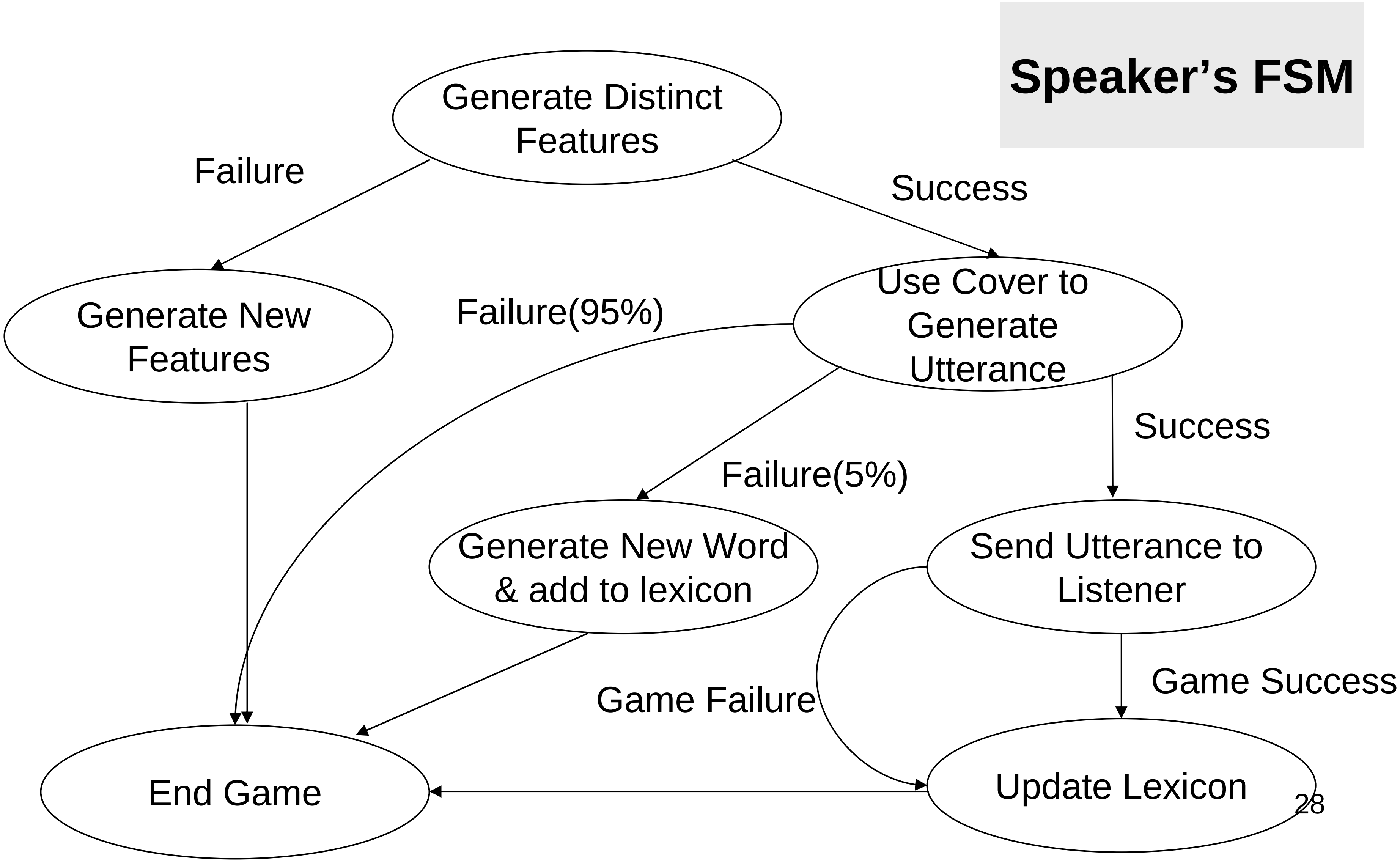
Language Games

- From the group of active agents we select a subset of agents called as ‘context’.
- Once a context is chosen we chose a ‘Speaker’ and a ‘Listener’ who are the active entities in these games.
- Another agent from the context is chosen as the topic of conversation.
- The speaker and the listener share the topic with each other through extra lingual means e.g. pointing, etc.

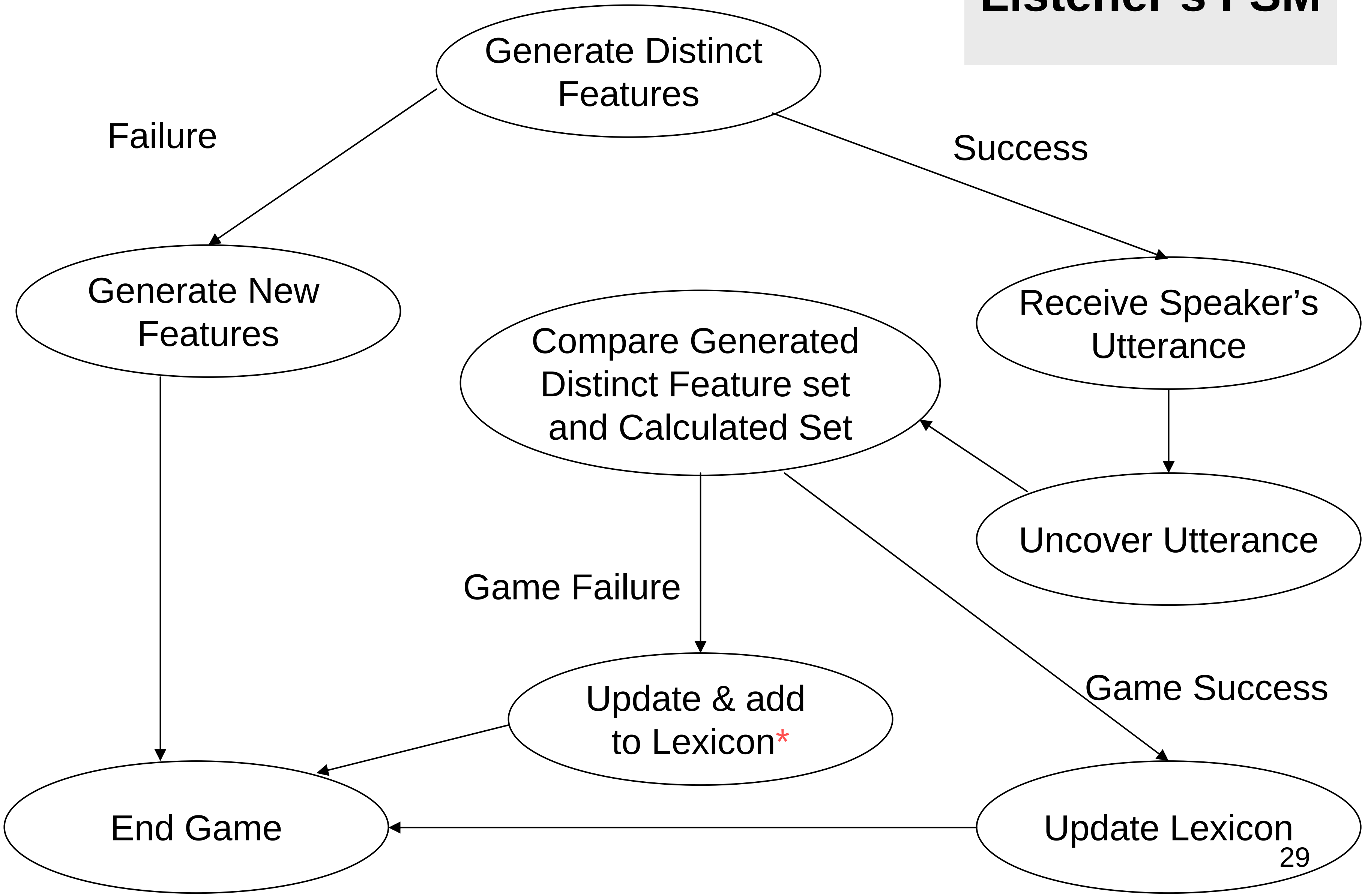
- The speaker and the listener try to come up with a distinct feature set for the topic.
- The game fails in the following cases
  – Speaker or listener do not have enough features to discriminate the topic from context.
  – Speaker does not have a word-feature mapping for the features in the distinct feature set.
  – Listener does not have enough feature-word mapping to generate a feature set for the utterance from speaker.

  - The distinct feature set generated by the listener is different from the feature set calculated by it from the utterance using the uncover function.
- The game succeeds only when the feature set generated by the listener and the feature set calculated from utterance of the speaker are overlapping.

Finite State Machines for Language Game



Listener's FSM



Simulation - Types

- One word utterances

  - This is the basic experiment.

- Multiple word utterances

  - New feature is introduced.

- Entrance of new agents

  - New words are created and sentences become more complex.

Simulation steps

- A speaker and hearer is randomly identified.
- Speaker selects the topic and shares the information with the hearer.
- Both speaker and hearer identify possible distinctive feature sets in topic.
- Speaker selects one set and translate it to words using cover function.
- The hearer interprets the utterance using uncover function and compares it with his expectations.

Language Game Possible Outputs

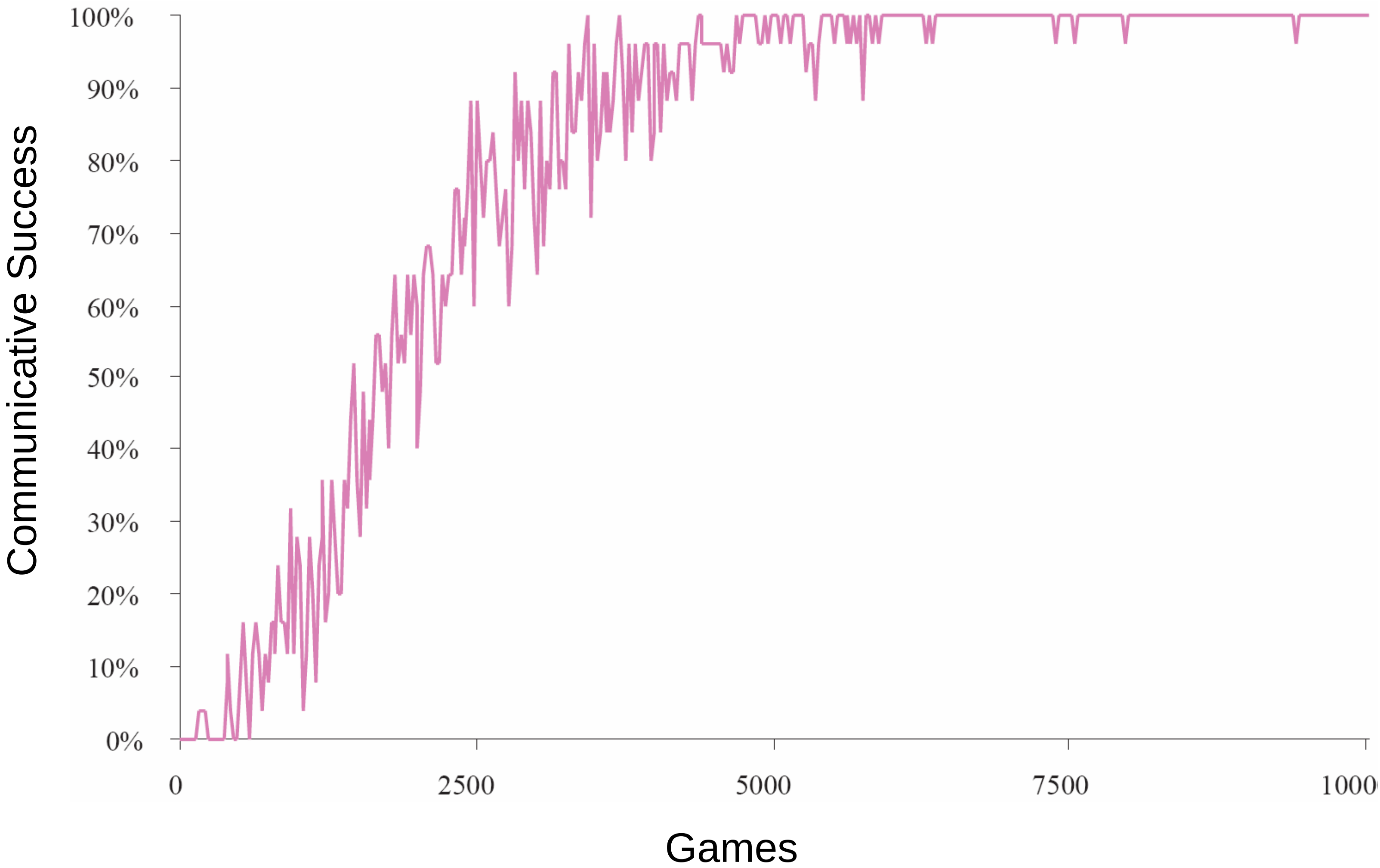
- No differentiation possible.
- The speaker does not have a word.
- The hearer does not have a word.
- The speaker and the hearer knows the word.

  - The meanings are compatible with the situation.

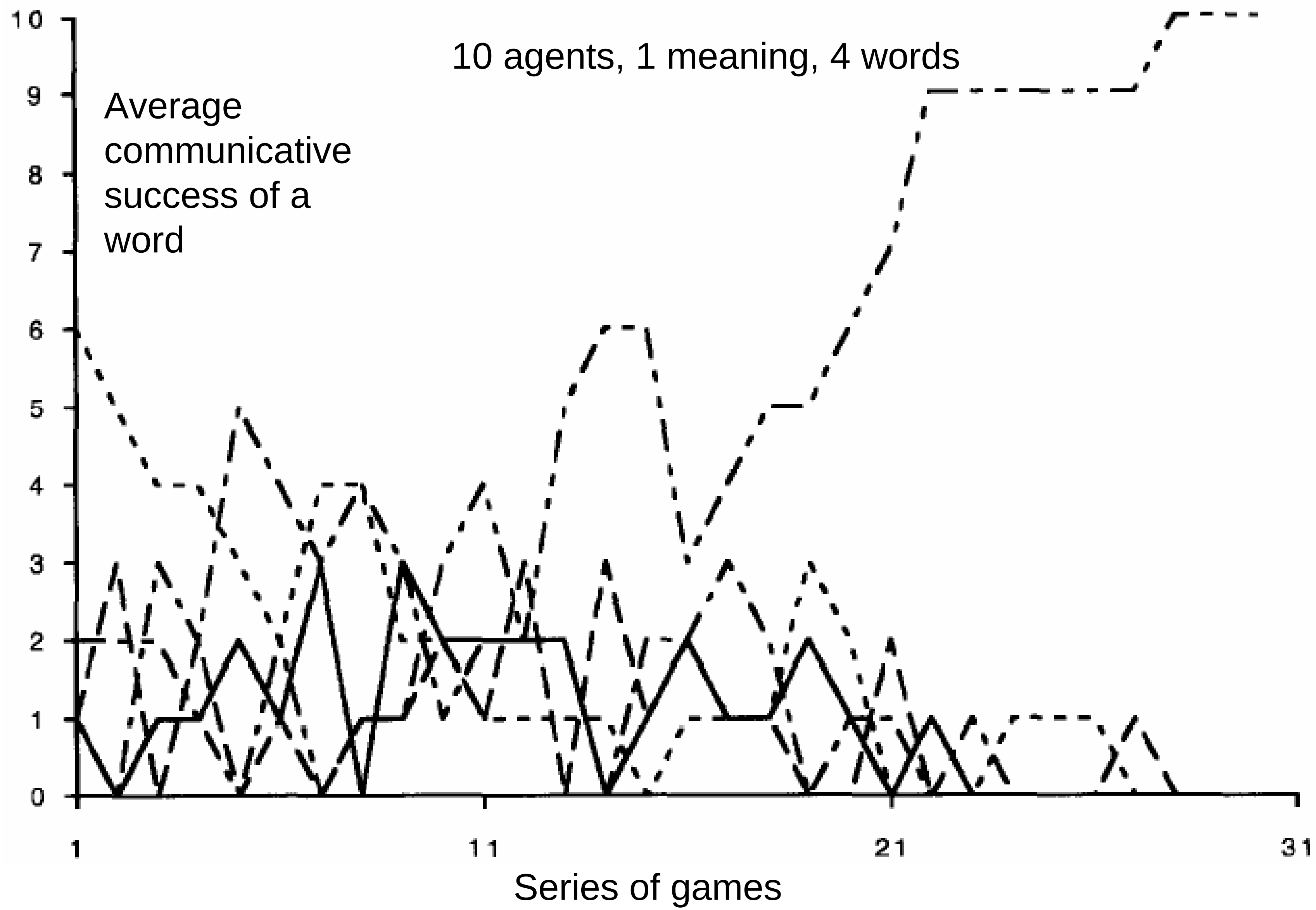
  - The meanings are not compatible.

Simulation Results (1)

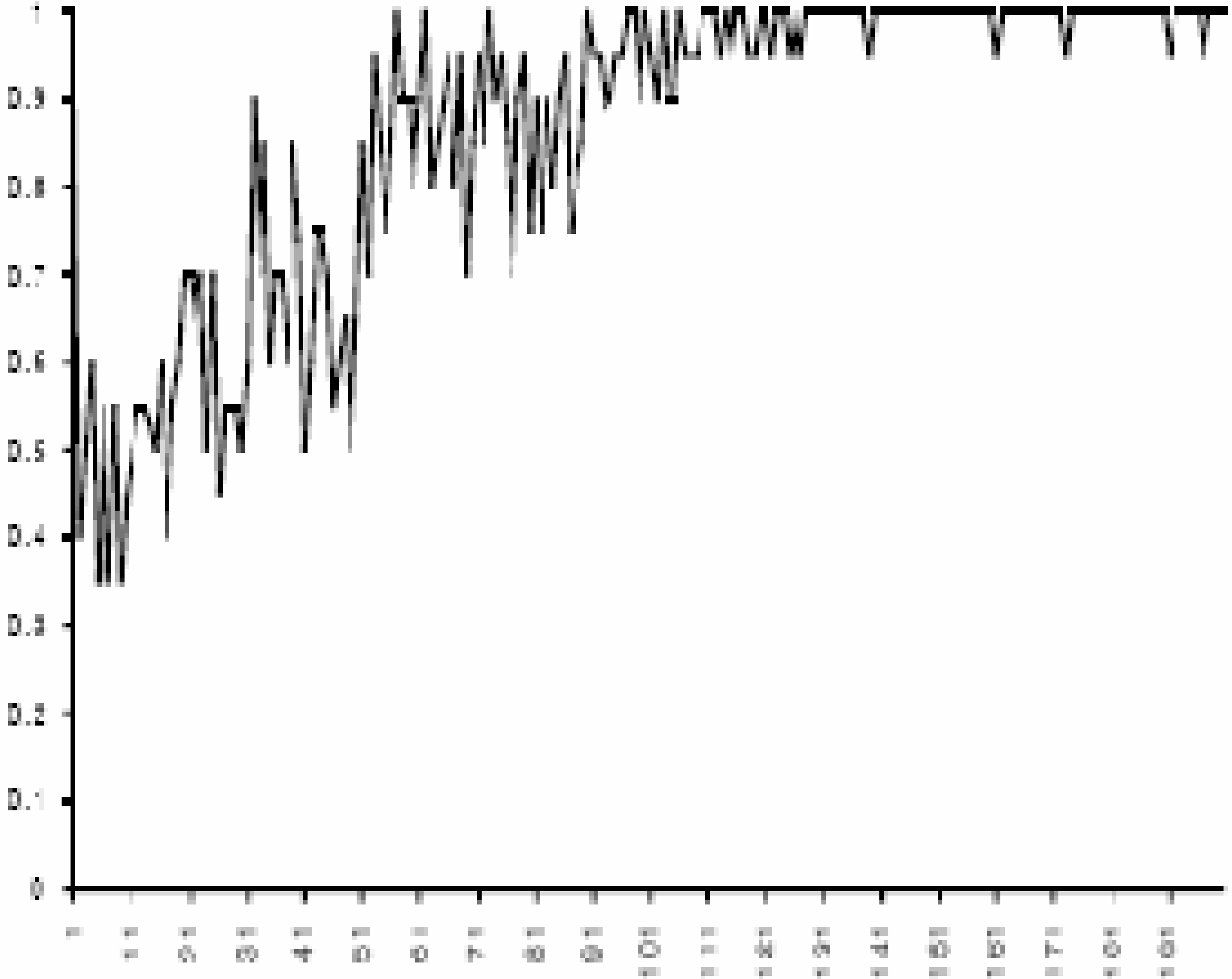
20 agents, 20 objects, software simulation



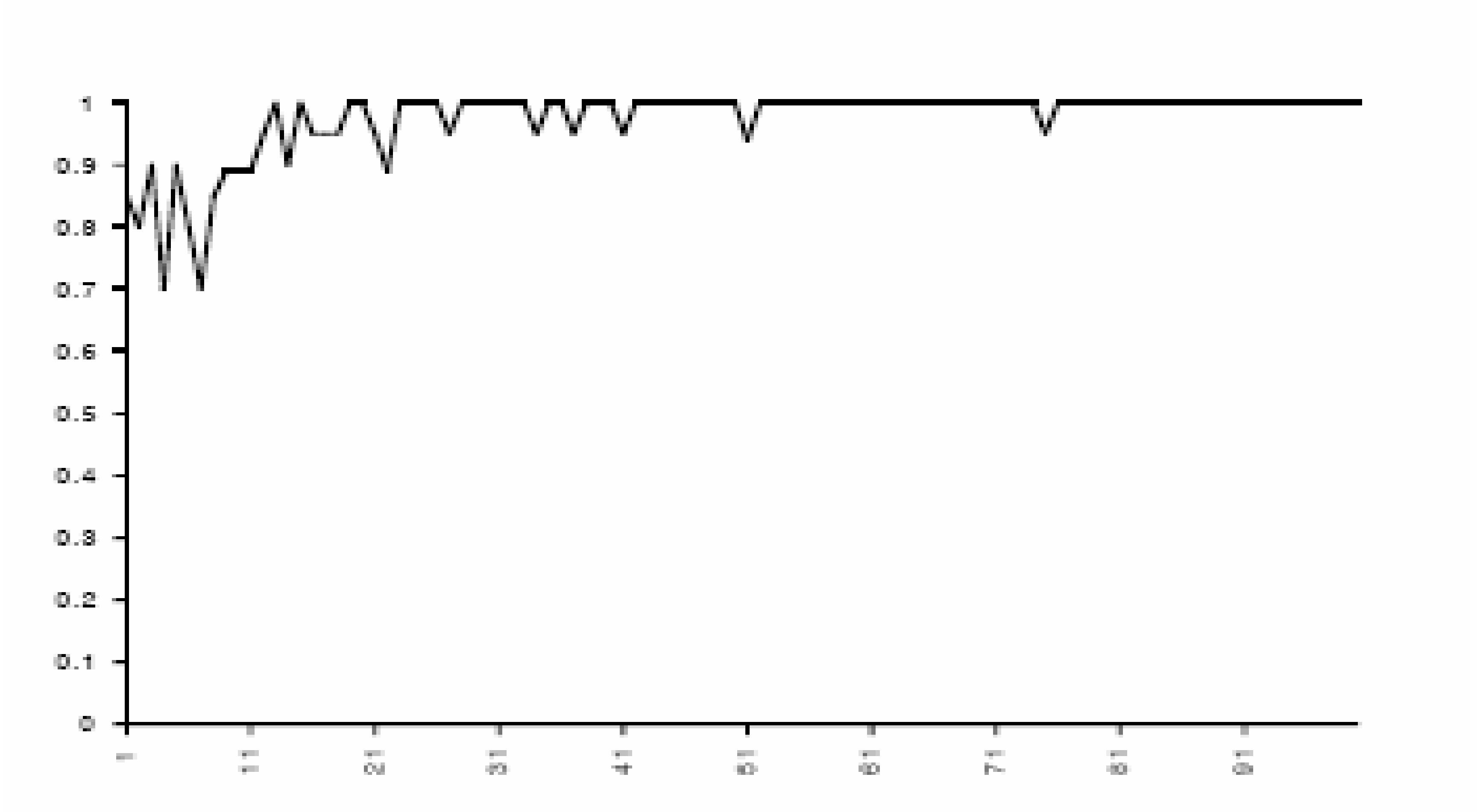
Simulation Results (2)



Simulation Results (3)



Simulation Results (4)



Conclusion

- Steels tries to model rudimentary language systems to study the evolution of language.
- The significant features of these experiments are:
  - Open system
  - Creation of new meanings
  - Only relevant meanings are lexicalized
  - Incoherence exists which is resolved at the time of finer disambiguation.
  - Meaning not necessarily identifiable from context

Conclusion (2)

  - Context plays a role in disambiguation of a given sentence.
  - Multiple sentence words emerge in order to disambiguate single words.
- The experiments highlight the following principles of language evolution
  - Complexity introduced progressively into the system
  - Adaptation
  - Selectionism
  - Self-organization

References

- Steels, L. (1996) Emergent Adaptive Lexicons. In: Maes, P. (ed.) (1996) Proceedigns of the Simulation of Adaptive Behavior Conference. The MIT Press, Cambridge Ma.
- Steels, L. (1996) The spontaneous self-organization of an adaptive language. Muggleton, S. (ed.) Machine Intelligence 15. Oxford University Press, Oxford.
- Steels, L. (1996) Perceptually grounded meaning creation. ICMAS 1996, Kyoto.
- Steels, L. (1997) Constructing and sharing perceptual distinctions. van Someren, M. and G. Widmer (eds.), 1997 Proceedings of the European Conference on Machine Learning (ECML), Springer-Verlag, Berlin.
- David DeAngelis(2005) The Origins of Syntax In Visually Grounded Robotic Agents