Introducing a Scene Building Game to Model Early First Language Acquisition

Kris Jack
Division of Applied Computing,
University of Dundee,
Dundee, Scotland, DD1 4HN.
kjack@computing.dundee.ac.uk

Abstract

This paper introduces a game which, when used in conjunction with a language learning algorithm, exhibits features of natural language production found to occur in young children. The game enables a rich and complex set of training data to be generated and acts as a quantifiable measure of linguistic ability, both for human and simulated players.

1 Introduction

Little is understood about the mechanisms involved in language acquisition that are responsible for transforming babbling children (6-8 months) into their linguistically adult level counterparts (48 months) (Ingram, 1989). The research described here introduces a scene building game that aims to implement an analogue of the curiosity-driven approach (Pinker, 1994) to natural language acquisition, that results in the ubiquitous linguistic stages.

A language acquisition algorithm is implemented that requires training data containing both linguistic and conceptual values. Although rich and varied bodies of text are readily available for academic usage, none contain conceptual data compatible with the algorithm. Such data must therefore be generated.

This paper suggests a method of data generation and presents results thus far as to the algorithm's success with the data. It is sectioned as follows; previous usage of the algorithm is revisited and the new problem is defined; a scene building game is described in detail and offered as a solution to the problem of incompatible data; the language learning algorithm is briefly described and its use of the new training data is considered, future research is described, and conclusions are drawn.

2 Previous Research

Previous research demonstrate the emergence of linguistic transition stages (Jack, Reed, and Waller, 2004). In particular, the transition from the one-word to the two-word stage is witnessed. Two production algorithms (one-word and two-word) take conceptual values as input (such as *square*, *circle*, *blue* or *red*) and produce the most likely string to partner it. Both one-word and two-word production algorithms compete from the outset. A clear trend of mostly one-word productions (absolute alignment driven) in early epochs is overtaken by two-word productions (syntactically derived alignment driven) in later stages, until two-word productions are absolute.

In previous research, the training data are small, unlike the data that a real language learner is exposed to. They are also clean, with no more than a limited number of adjective-noun word co-occurrences contained within string input. To test the effectiveness of the production algorithm when faced with a more realistic set of data, the data must contain;

1. A larger vocabulary.
2. Longer strings.
3. A greater linguistic complexity.
4. Noisy elements.

Furthermore, compatible data is essential. Conceptual values must be embedded within the data. Such data do not exist in academic resources.

5. Conceptual values.

This paper focuses on a solution that satisfies these five requirements.

3 A Scene Building Game to Encourage Natural Language Generation

Games have long been a popular tool employed by psycholinguists in determining linguistic properties in children. More recently, computational linguists have harnessed the power of games in the study of language acquisition (Oliphant and Batali, 1997; Kirby, 1999; Steels and Kaplan, 2001). A new game is now introduced that encourages the generation of natural language along with appropriate conceptual data. The nature of the game also allows for a quantifiable success/failure metric to determine a player's linguistic ability, within the constraints imposed in the game.

3.1 Experimental Setup

A game is played with two players, a speaker and a listener. The speaker is the only player who is allowed to talk. The speaker must describe the ongoing generation of a scene and the listener must generate a new scene based on that description. The more alike the two scenes are at completion, the better the communication between the players.

One player takes the role of the speaker, while the other is the listener. Both speaker and listener look at their own scenes (whiteboards) and are unable to see the other's. Both scenes are blank from the outset. A change is made to the speaker’s scene.

Currently, all changes are in the form of a shape being added to the scene. There is therefore a wide scope for future, more complex, versions of the game that could include shape removal, shape repositioning, colour changes, size changes, actions such as moving, bouncing, etc. There are also a limited number of shapes that can be added to the scene (circles, squares and triangles that are either red, blue or yellow). All shapes are the same size. The game has been designed to allow for many variable features although only two are varied in this research (shape and colour).

The speaker is asked to describe the change that is made to the scene so that someone else can reconstruct it based on their description alone.

3.2 Playing the Game

The scene building game is played as an analogue to the curiosity-driven approach that children demonstrate but with the implicit questioning. When a change is made to the scene, the speaker is expected to describe it, without being prompted to. The listener then makes the change to their own scene, using only the speaker’s instruction.

Another change is made to the speaker’s scene and the routine of describing a change followed by making that change continues until the complete scene has been rebuilt by the listener (scenes, including changes, have been defined beforehand).

By adding more than one listener to the game, many listener’s scenes can be compared to the speaker’s scene. On completion, the speaker’s scene is compared to each of the listener’s scenes individually. A quantifiable metric exists that defines the similarity of the two scenes based on object positioning. The more alike a listener’s scene is to the speaker’s, the better the communication between that pair. The best listener (for that speaker) can therefore be determined based on the comparison made.

3.3 Using the Game To Generate Natural Language

If the game is transferred to a computer system (screens instead of whiteboards, typing instead of speaking, reading instead of listening, simulated change instead of physical change) then a wealth of natural language and conceptual values can be recorded as games are played. Conceptual values are based upon changes that have been pre-programmed in the scene. For example, if a green circle appears on the screen then the concepts of *green* and *circle* will be used as conceptual input. The concepts are not related in any way (for example, colour or shape categories). In the current version of implementation, these conceptual values are directly fed into the system’s input channel, but a simulated version could also be constructed, allowing a camera to detect shape introductions to a scene, and extract the colour property from the shape (Howell, Becker, and Jankowicz, 2001).

Natural language is generated by recording the speaker’s input. In this case, the speaker will type a description of the scene change and it shall be recorded along with the corresponding conceptual values. No restrictions are placed upon the size or complexity of the input string.

Playing the game generates two types of training data. Training data are comprised of string to concept relationships. The first type of data are concept-driven. These are derived from the speaker’s string inputs corresponding to conceptual variations in the scene.

The second type of data are string-driven. They are formed by combining listener’s scene changes (that correspond to conceptual values) with string output (originally generated by the speaker).

3.4 Determining Linguistic Ability

A human player can be replaced by a computer player. Assuming that the computer player takes on the role of a listener, then its scene production can be compared to a human player’s scene productions. Upon consistently producing better than average results, compared to human listeners, the computer system demonstrates a better than average linguistic aptitude (similar to metric employed in Turing Test (Turing, 1950)).

The resulting measure is therefore a comparative indication of the player’s linguistic aptitude, and cannot be used in isolation. This research is ongoing and a complete version allowing the computer to create its own scenes has not yet been implemented.

4 Training Data

As previously stated, the focus of this research is data generation. To date, four hundred string-concept relationships have been generated. They were generated by four people playing the scene building game. Each player took on the role of the speaker and, sequentially and in isolation, described the construction of ten scenes, each containing ten scene changes. No listener trials have yet been conducted.

4.1 Suitability of Training Data

The training data have been examined with respect to the requirements previously imposed on them.

1. A larger vocabulary.

Previously, X different words were in use.

The game generated Y different words.

2. Longer strings.

Previously, the average length of a string was 1.5 words.

The game generates an average string length of 8.3 words.

3. A greater linguistic complexity.

Previously, one-word descriptions containing a noun or adjective and two-word descriptions containing an adjective-noun word co-occurrence were used.

The game generates nouns, verbs, adjectives, pronouns, adverbs, prepositions, conjunctions, articles, false starts and anaphora. The structural combinations are complex.

4. Noisy elements.

Previously, all strings were correct descriptions of their concepts.

The game generates correct and incorrect statements (in some cases irrelevant).

5. Conceptual values.

It is essential that all strings are accompanied by conceptual values. The game forms these relationships.

The game therefore succeeds in meeting the requirements for training data generation.

4.2 Further Discussion of Training Data

Although the rules of the game are simple, the training data generated are complex and perfectly suited to the language learning algorithm. Many training sets can be generated in a short period of time and with few participants. Analysis of the training data reveals further areas of interest.

4.2.1 Description Styles

Each player is given the same instructions. A distinctive style can be observed in each player's descriptions. They vary considerably, yet are all acceptable descriptions of the same scene change (in this case, the introduction of a blue circle):

1. "A blue circle has appeared to the left of the red square"
2. "a blue circle to left of the red square"
3. "Blue circle: to the left of the red square"
4. "a blue circle about 1cm left of the square"

The algorithm must be robust enough to associate different styles of descriptions with the same conceptual values.

4.2.2 Noisy Data

The data contains noisy elements. Among these were spelling mistakes, grammatical errors, different writing styles, and incorrect textual descriptions of scene changes. Noise is kept in the training data to observe the effects on algorithmic performance.

4.2.3 Spoken Versus Textual Data

There was concern that the translation from the original game (non-computer based) to the new game (computer based) may impact upon the kind of language generated. In particular, there was concern that textual descriptions would differ too much from spoken descriptions. Two versions of the game have been implemented and, as suspected, textual and spoken descriptions do differ. One player described the same scene change vocally as –

"That's a red circle in the top right hand corner of the screen"

and textually as –

"a red circle in top right hand corner of screen"

In terms of implementation, there is a transcription overhead involved in the spoken version that does not exist in the written version. Transcription also introduces the opportunity for human error. In terms of grammatical complexity, the textual version is just as complex as the spoken version, although contains cleaner data (e.g. less "ums" and "ers"). The loss in training data complexity is therefore negligable and the time saving benefits are enourmous. All future implementations will use only the textual version.

5 Language Learning Algorithm

The key functions of the language learning algorithm are to assimilate data and construct description pairs (string and concept values) based on either a string or set of concept values (comprehension and production). Since the game has not been tested for comprehension (listener task), only assimilation and production shall be considered here.

5.1 Design

5.1.1 Assimilation

All learning occurs during assimilation. Training data are entered at this stage and are always of the form of a string and concept values. A form of perfect alignment based learning (van Zaanen, 2000) is used, where alignment occurs between string and concept value relationships.

```
if string and concept values in system
        increment occurrence frequency of pairing
else
        add pairing to system
        for all pairings in system
                if new pairing is similar to existing pairing
                        add equalities to system
                        add inequalities to system
```

Algorithm 1: Assimilation of Training Data

Relationships are recorded between string and concept values. All combinations of substrings, and permutations of concepts values are produced. The maximum number of new relationships that can be produced from the entry of a string containing **x** words and a concept containing **y** values is therefore:

$$(\mathbf{x} * (\mathbf{x} + 1) * (\mathbf{y} * (\mathbf{y} + 1) + 1)) / 2.$$

The minimum number of new relationships that can be produced is always zero, as the relationship may already exist. Clearly, the system is required to cope with a large number of relationships when the average number of words per sentence is 8.3 words (found in training data).

5.1.2 Production

Production is simulated by finding the most appropriate string to relate to a given set of concept values.

```
find all group$_{variant}$ that contain the concept values
for each group$_{variant}$ find all group$_{substitute}$
build all possible groups from group$_{variants}$ and group$_{substitutes}$
sum results in agreement
```

Algorithm 2: Production from Concept Values

As mentioned previously, one-word grammar and two-word grammar compete in the production stage. The system does not distinguish between the grammars, following the above algorithm with precision. One-word production is defined as a string to concept relationship that does not use group$_{substitute}$ data; two-word production uses group$_{substitute}$ data.

One-word production employs a shallow perfect alignment model, whereas two-word production employs a deep alignment model.

5.2 Use of Training Data

The game allows for the generation of training data that reflect natural language input more accurately than previously hand-written data. These data are much larger in size and, as a result, the efficiency of the system suffers. Both assimilation and production times increase considerably. Note that the algorithm has not been adapted since previous testing, so efficiency problems are expected.

Results thus far indicate that the algorithm demonstrates the same maturation using the new training data as is seen using the previous data. The algorithm, as expected, produces sentence descriptions comparable to the one-word and two-word stages observed in children, reducing training data such as “A red square has appeared in the top right of the screen” to sentences like “red” and later “red square”. Such sentence reductions are common in early childhood (Bloom, 1973). The algorithm produces the effect of extracting the salient substrings from a large volume of data.

6 Future Research

Future versions of the game shall expand the system’s conceptual framework. A greater number of conceptual values will allow the system to discriminate features such as relative and absolute shape positioning. Trials shall be run with players who take on the role of the listener and quantifiable metrics shall be further explored. The algorithm shall be recoded to improve efficiency.

7 Conclusion

The scene building game encourages the generation of complex linguistic data that contain examples of difficult problems for computational linguists. The data generated is perfectly suited for use in the language learning algorithm that demonstrates the emergence of linguistic stages as witnessed in child language acquisition. Inherint to the game is a quantifiable metric allowing for human and simulated players to be rated for linguistic aptitude.

8 Acknowledgements

The first author is sponsored by a studentship from the EPSRC.

Thanks to excellent guidance in the Division of Applied Computing.

References

L. Bloom, 1973. One Word at a Time. The use of single-word utterances before syntax The Hague, Mouton.

S.R. Howell, S. Becker, and D. Jankowicz, 2001. *Modelling Language Acquisition: Lexical Grounding Through Perceptual Features*. In Proceedings of the 2001 Workshop on Developmental Embodied Cognition

D. Ingram, 1989. *First Language Acquisition. Method, Description and Explanation*. Cambridge: Cambridge University Press.

K. Jack, C. Reed, and A. Waller, 2004. A Computational Model of Emergent Simple Syntax: Supporting the Natural Transition from the One-Word Stage to the Two-Word Stage. In *Proceedings of the Workshop Psycho-Computational Models of Human Language Acquisition (COLING 2004)*, pages 61-68.

S. Kirby, 1999. Syntax out of learning: The cultural evolution of structured communication in a population of induction algorithms. *In Proceedings of ECAL99 European Conference on Artificial Life, D. Floreano et al. ed.* pages 694-703, Berlin: Springer-Verlag,

M. Oliphant and J. Batali 1997. Learning and the emergence of coordinated communication. *Centre for Research in Language Newsletter*, 11(1).

S. Pinker, 1994. *The Language Instinct. The New Science of Language and Mind*. Allen Lane, Penguin Press.

L. Steels and F. Kaplan, 2001. AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, vol. 4(1):3-32. John Benjamin’s Publishing Company, Amsterdam, Holland.

A.M. Turing, 1950. Computing Machinery and Intelligence. *From Mind LIX, no. 2236, Oct. 1950* : 433-460

M. van Zaanen, 2000. Learning structure using alignment based learning. In *Proceedings of the Third Annual Doctoral Research Colloquium (CLUK)*, pages 75-82.