From Syllables to Syntax: Investigating Staged Linguistic Development through Computational Modeling

Kris Jack (kjack@computing.dundee.ac.uk)
Chris Reed (chris@computing.dundee.ac.uk)
Annalu Waller (awaller@computing.dundee.ac.uk)
Applied Computing, University of Dundee,
Dundee, DD1 4HN, Scotland.

Abstract

A new model of early language acquisition is introduced. The model demonstrates the staged emergence of lexical and syntactic acquisition. For a period, no linguistic activity is present. The emergence of first words signals the onset of the holophrastic stage that continues to mature without syntactic activity. Syntactic awareness eventually emerges as the result of multiple lexically-based insights. No mechanistic triggers are employed throughout development.

Keywords: Computational modeling; Emergence of Syntax; Item-based Learning; Language Acquisition.

Introduction

Children acquire language in stages, first learning words and later showing sensitivity to their syntactic properties. Processes that demonstrate distinct behaviors at different stages of development are difficult to model within a unified system. As a result, lexical and syntactic processes are often modeled independently from one another. Bridging the gap between these models will increase understanding of the behavioral shift that ushers in syntactic awareness.

Background

Modeling Word-to-meaning Mappings

Children learn the meanings of a small number of words early in linguistic development. These first words are often non-formulaic (Wray, 2002). A non-formulaic word expresses a word-to-meaning relationship that is not a function of the word's internal parts.

Siskind (1996) investigates word-to-meaning mappings using cross-situational analysis. Cross-situational analysis takes advantage of word-meaning co-occurrences to establish relationships. His simulations show considerable success, offering a robust solution to the problem under a variety of circumstances. Steels (2001) considers the problem of establishing such mappings through language games. Treating language as a complex adaptive system, he shows that social pressures to communicate, through games, encourage the development of a self-organized lexicon. Lexical acquisition is also studied within a developmental framework. Regier (2005) shows that interesting lexical phenomena, such as fast-mapping, can arise without internal mechanistic changes. Attentional learning plays an important role in language acquisition.

Modeling the Emergence of Syntax

All natural languages employ syntax. Syntax allows individuals to both understand and produce novel utterances. Unlike non-formulaic language, syntactically produced utterances are a function of their internal parts.

Elman (1993) finds that simple and complex linguistic structures can be learned by a neural network, but only if the former are acquired before the latter. To ensure simple structures are learned first, the neural network's memory length is initially small, and increased during training. This 'maturational' growth allows both types to be acquired without staged input. Dominey and Boucher (2005) investigate developmental phenomena within a grounded robot. A form of syntactic bootstrapping arises as grounded <sentence, event> pairs are learned. The model, however, employs a manual trigger that activates the syntactic component, an inadequate explanation for the emergence of syntax. Kirby (2001) considers language transmission from generation to generation through the Iterated Learning Model. He demonstrates that transmission bottlenecks, that determine the amount of linguistic exposure a learner receives, have an important effect on the emergence of syntax. The bottleneck can be neither too narrow nor too wide for syntactic structures to be derived.

Bridging the Gap between Words and Syntax

None of these models show the developmental shift from lexical to syntactic awareness reflected in child language development. Jack, Reed and Waller (2004) consider the transition from the one-word stage to the two-word stage. A model is trained on <string, meaning> pairs, testing interpretation of strings at each training epoch. In early training, a preference for non-formulaic (lexical) interpretation emerges. As training continues, this preference fades, giving way to formulaic (syntactic) interpretations. The behavioral change is an emergent property of the training process and not artificially triggered. Although a developmental shift is witnessed it appears very early in the model and the purely lexical period is very short, unreflective of natural child language development.

Modeling the Developmental Shift

Children do not understand syntactically complex utterances from birth. First words, produced at around 10-months-old (Bates & Goodman, 1999), are non-formulaic, with no indication of syntactic properties. By around 18-months-old, syntactic awareness emerges (MacWhinney & Bates, 1989). An accurate model of language acquisition should reflect the development from the holophrastic stage (non-formulaic) to the early multi-word stage (formulaic).

The Holophrastic Stage Specification

During the holophrastic stage, the model shows no syntactic awareness. All successful string-to-meaning mappings are performed through non-formulaic interpretation i.e. given the string “all gone”, the appropriate meaning is mapped directly without reducing the string to its individual parts, “all” and “gone”.

The Early Multi-word Stage Specification

During the early multi-word stage, the model shows syntactic awareness. Some successful string-to-meaning mappings are performed through formulaic interpretation i.e. given the string “all gone”, it is reduced to its individual parts, “all” and “gone”. Non-formulaic language persists.

A symbolic model is implemented to investigate this developmental shift. The remainder of the paper describes this model and discusses its behavior.

The Model

Training Data

The Miniature Language Acquisition framework (Feldman, Lakoff, Stolcke, & Weber, 1990) allows language acquisition to be studied by coupling visual events with linguistic descriptions. Under this framework, a scene building game is played. An object appears in a scene and is described. The object always appears next to another object. These <event, description> pairs are entered into the model as training data.

Objects are expressed by a set of feature tuples. A feature tuple expresses a value and an object identifier. Values are derived from simulated visual data, consistent with computer vision technology capabilities. Object identifiers uniquely identify the object that the value belongs to. Since there are always two objects in an event, they are numbered 1 and 2. 1 is the first object in the scene while 2 is the second. Objects vary in shape, color and position. The object {<*red*, (1)>, <*circle*, (1)>} reflects that the first object in the scene is a red circle. Object identification is present in infants (Kellman, Gleitman, & Spelke, 1987).

Events are expressed by a set of feature tuples comprising two objects and the relationship between them. The event {<*red*, (1)>, <*circle*, (1)>, <*pink*, (2)>, <*cross*, (2)>, <*above*, (0)>, <*right*, (0)>} reflects that a pink cross appeared to the upper right of a red circle. Relative positions are expressed as binary relationships along horizontal and vertical planes, as suggested by infant interpretations of spatial locations (Quinn, 2003).

Descriptions are syllable-segmented strings. Descriptions are not word-segmented as fluent speech contains no known acoustic analog of the blank spaces in text (Brent & Siskind, 2001). A syllabic base is implemented as infants are likely to represent sound based on a syllable covariant (Dehaene-Lambertz & Houston, 1998; Mehler, Dupoux, Nazzi, & Dehaene-Lambertz, 1996). Word spellings are retained for readability unless words share syllables e.g. low occurs in lower and yellow, producing “low er” and “ ye low”.

Training data are randomly generated. Objects can have one of 10 colors and 10 shapes, allowing 100 objects. An object can appear in one of eight relative locations to one another. This allows a total of 80,000 unique events (100 objects x eight relative locations x 100 objects). Descriptions are generated through a grammar specification (Table 1). The grammar is instantiated when producing training data alone and is not accessible by the model during learning. The grammar is supplied for reader's convenience.

Table 1: The grammar specification for event descriptions.

S = NP1 REL NP2	NP1 = a COLOR SHAPE
NP2 = the COLOR SHAPE	REL = REL1 | REL2
REL1 = a bove | be low | to the REL4	REL2 = REL3 REL4
REL3 = to the u pper | to the low er	REL4 = right of | left of
COLOR = red | blue | pink | green | white | black | ye low | gray | lime | pur ple	
SHAPE = cir cle | dia mond | heart | cross | tri ang gle | star | rec tang gle | square | pen ta gon | hex a gon	

Overview

The model is designed to investigate the appearance of lexical and syntactic sensitivity. It is implemented as a symbolic system. A set of training data (<event, description> pairs) are randomly generated and entered into the model. Each pair is analyzed by the Lexical Analysis Unit. Lexical items are determined from data regularities through cross-situational analysis (Siskind, 1996). These items are processed by the Syntactic Analysis Unit that derives syntactic rules and phrasal categories. Syntactic rules specify the interaction between phrasal categories.

The Lexical Analysis Unit

Training data are entered into the model in the form of <event, description> pairs. Lexical items are derived based on these data. Given that strings are syllable-based, word boundaries are not provided and must be derived. In some cases, these word boundaries overlap, increasing ambiguity. Meaning 'boundaries' must also be derived since not all feature tuple sets are singletons e.g. *below* can be represented as {<*below*, (0)>, <*even_horizontal*, (0)>}. The model must further derive how these strings and meanings are related to one another.

The learning algorithm is best described by way of example. The model contains pair (1). On the entry of pair (2), the model checks if the pair has been encountered before. If so, then a count is kept of the number of times that it has appeared and lexical analysis ends. If not, then a form of cross-situational analysis begins to identify event and string equalities. It is assumed that words will co-occur more often with their referents than with other meanings. Regularities are extracted across events and descriptions individually before recombining the results.

1. <{<*red*, (1)>, <*circle*, (1)>, <*pink*, (2)>, <*cross*, (2)>, <*above*, (0)>, <*right*, (0)>},
   "a pink cross to the u pper right of the red cir cle">
2. <{<*green*, (1)>, <*cir cle*, (1)>, <*red*, (2)>, <*diamond*, (2) >, <*even_vertical*, (0)>, <*right*, (0)>},
   "a red dia mond to the right of the green cir cle">

Event regularities are derived based on feature tuple equality. Feature tuple comparisons are value sensitive and identifier insensitive. That is, the feature tuple <*red*, (1)> is equal to any feature tuple with the value *red* regardless of identifier value. All feature tuple equalities are extracted over the two events, producing (3) and (4).

3. {<*red*, (1)>, <*circle*, (1)>, <*right*, (0)>}
4. {<*cir cle*, (1)>, <*red*, (2)>, <*right*, (0)>}

Description comparisons are syllable form sensitive reflecting infants' sensitivity to syllabic patterns (Houston, Santelmann, & Jusczyk, 2004). Descriptions are aligned, (5) and (6), and syllable lists are extracted, producing (7) and (8).

5. "a pink cross to the u pper right of the red cir cle"
6. "a red dia mond to the right of the green cir cle"

7. "a", "to the", "right of the", "red", and "cir cle"
8. "a", "red", "to the", "right of the", and "cir cle"

Event and description regularities are recombined producing <{feature tuple}, string> pairs. All combinations of regularities from the first event and the first description produce some of co-occurrences (e.g. <{<*red*, (1)>, <*circle*, (1)>, <*right*, (0)>}, "a">), while second event and second description combinations produce the remainder. Each pair is re-entered into the model and activates the same process as the original training data.

Often, more than one {feature tuple} accompanies each string after learning. To avoid ambiguity, each string must be represented by only one {feature tuple}. Children actively avoid synonymy during language learning, following a principle of mutual exclusivity (Markman & Wachtel, 1988). Given the list of {feature tuple}s to which a string is related, the {feature tuple} with the closest distribution to the string is selected. In some cases, a string may be represented by two {feature tuple}s that are equal. For example, <{<*red*, (1)>}, "red"> means that "red" is associated with the redness of object 1 and <{<*red*, (2)>}, "red"> means that "red" is associated with the redness of object 2. These relationships are combined and written as < {<*red*, (1, 2)>}, "red">, representing the redness of either object 1 or 2.

Each <{feature tuple}, string> pair indicates a syllable set-to-meaning relationship. If more than one string is related to the same {feature tuple} then synonymy occurs. Synonymy is rare in natural language. The string with the highest probability of being represented by each unique {feature tuple} is selected. The most probable <{feature tuple}, string> pairs are stored as *lexical items* in the model. These items are not always representative of adult word-to-meaning boundaries. Learning phenomena such as under-generalization and mismatching are encountered. For example, the word "red" should be representative of redness in any object but is sometimes under-generalized to a single one object. Mismatches such as <{<*circle*, (1)>}, "to the"> are also found. These phenomena are indicative of the holophrastic stage in learning.

The Syntactic Analysis Unit

Non adult-like lexical items can also express syntactic relationships. Lexical item (9) is a formulaic function of lexical items (10) and (11). The Syntactic Analysis Unit is responsible for discovering and encoding this relationship.

9. <{<*red*, (1, 2)>, <*circle*, (1, 2)>}, "red cir cle">
10. <{<*red*, (1, 2)>}, "red">
11. <{<*circle*, (1, 2)>}, "cir cle">

Syntactic relationships are discovered within lexical item triples (such as (9)-(11)). One lexical item, (9), must be the function of the two others items, (10) and (11). The lexical items must satisfy both string and {feature tuple} relationships. Given two strings, the model must produce the third through string concatenation, i.e. $\text{string}_1 + \text{string}_2 = \text{string}_3$. Also, given two {feature tuple}s, the model must produce the third through set union i.e. $\{\text{feature tuple}\}_1 \cup \{\text{feature tuple}\}_2 = \{\text{feature tuple}\}_3$. {Feature tuple} equality is identifier insensitive, so identifiers need not match.

Rules capture these relationships. They relate Phrasal Categories (PCs) to one another by the application of Transformations (Ts). Each new term is defined before the rule is presented.

Rules are expressed in the form $PC_1 = PC_2(T_1)\ PC_3(T_2)$, where PC_1 is produced by combining the results of PC_2, being transformed by T_1, and P_3, being transformed by T_2.

Phrasal Categories are expressed as the pairing of a set of strings and a list of feature tuple identifiers, <{string}, (identifier)>. PCs are created to support rule relationships. There are two kinds of PCs; parent and child. Given the rule $PC_1 = PC_2(T_1)\ PC_3(T_2)$, PC_1 is a root, while PC_2 and PC_3 are children. Root PCs acquire lexical item 1's data and identifier end points from T_1 and T_2. Child PCs are populated with strings from the original lexical items that they are derived and the appropriate T start point.

Transformations are expressed as a set of feature tuple identifier pairs, {feature tuple identifier pair}. Feature tuple identifier pairs define the mapping from a start point to an end point, in transforming feature tuple identifiers, <start identifier, end identifier>.

The Syntactic Analysis unit produces rule (12) from lexical items (9)-(11).

12. $PC_1 = PC_2(T_1)\ PC_3(T_2)$, where
    PC_1 = <{"red cir cle"}, ((1, 2), (1, 2))>,
    PC_2 = <{"red"}, ((1, 2))>,
    PC_3 = <{"cir cle"}, ((1, 2))>,
    T_1 = {<(1, 2), (1, 2)>} and T_2 = {<(1, 2), (1, 2)>}.

Rule (12) expresses a functional path to derive lexical item (9), using items (10) and (11). It specifies the mapping from the meaning of items (10) and (11) to producing item (9). Rule (12) shows how to generate a {feature tuple} that represents the string “red cir cle”. First, the model searches for lexical items that represent the child PCs. Lexical items for “red” and “cir cle” are found; <{<*red*, (1, 2)>}, “red”> and <{<*circle*, (1, 2)>}, “cir cle”> respectively. Each lexical item is transformed based on its PC's T. The lexical item for “red” is transformed by T_1 and “cir cle” by T_2. In this case <{<*red*, (1, 2)>}, “red”> becomes <{<*red*, (1, 2) >}, “red”> (no change) and <{<*circle*, (1, 2)>}, “cir cle”> becomes <{<*circle*, (1, 2)>}, “cir cle”> (no change). The results are joined together through set union producing < {<*red*, (1, 2)>, <*circle*, (1, 2)>}, “red cir cle”>.

The Syntactic Analysis Unit analyzes every combination of lexical item triples and produces a rule for each group that expresses a syntactic relationship. Rules can express similar relationships. Rules (13)-(15) all express the same relationship. Rule (13) is the short-hand version of rule (12) for improved readability. They state, that “red cir cle”, “blue cir cle” and “pink dia mond” can each be produced by applying the same transformation rules to their children. A transformation rule must have the same start point and end point to be considered equal.

13. {“red”}((1, 2) -> (1, 2)), {“cir cle”}((1, 2) -> (1, 2))
14. {“blue”}((1, 2) -> (1, 2)), {“cir cle”}((1, 2) -> (1, 2))
15. {“pink”}((1, 2) -> (1, 2)), {“dia mond”}((1, 2) -> (1, 2))

When rules are found to express the same relationship, they are merged together. Merging rules (13)-(15) produces (16). (16) has the generative capacity to produce 6 different strings; “red cir cle”, “blue cir cle”, “pink cir cle”, “red dia mond”, “blue dia mond”, and “pink dia mond”.

16. {“red”, “blue”, “pink”}((1, 2) -> (1, 2)), {“cir cle”, “dia mond”}((1, 2) -> (1, 2))

Rule (16) captures the English grammar rule, NP = Adj. N, where the 'adjective' set contains “red”, “blue”, and “pink” and the noun set contains “cir cle” and “dia mond”. The rule states, among other combinations, that when the string “red” directly precedes the string “dia mond”, a *red diamond* is being indicated. To emphasize, the rule does not just indicate that there is redness in the scene, nor that there is diamond in the scene, but that there is an object in the scene that shares both the properties *red* and *diamond*.

From syllable segmented strings combined with feature based meanings, English-like grammar rules are derived. Each rule defines a mapping based not only on individual lexical items, but groups of lexical items, or PCs, producing syntactic units. These lexical items are established by drawing word and meaning boundaries. The PCs are established by drawing lexical item boundaries. The fixing of these lexical item boundaries allows the model to treat different words in a similar way and, ultimately, produce novel relationships such “red dia mond” in the previous example. Furthermore, the lexical item boundaries change the model's perception of lexical status. While lexical analysis produced items such as “red cir cle”, syntactic analysis draws a boundary through the string and its related meaning, allowing it to be deconstructed and reconstructed with the application of other items. PC role (parent or child) and membership, therefore, is a better indicator of lexical status than the lexical items themselves.

Comprehension

The model is tested for evidence of language acquisition through comprehension tasks. Given a string, the model must derive a {feature tuple}. Following the example from the last section, assume that the model contains rule (16) and has never encountered the string “red dia mond” in training.

PC membership offers a better indication of lexical status than lexical items. The model searches for the string in all PCs. If the string appears in a PC then its lexical item representation is retrieved. If the string does not appear in a PC then the comprehension process continues regardless. In this case, the model has never encountered the string “red dia mond”, so it not a member in any PC.

The model contains rules that specify how to produce meanings for a number of strings. These rules take two substrings as input. Using these rules, the string to parse is dissected into two parts. Any string that contains more than one syllable can be dissected. The string “red dia mond” is dissected, by syllable boundaries, producing the pairs <“red”, “dia mond”> and <“red dia”, “mond”>. Each string is recursively processed by the comprehension algorithm detailed in this section. Taking <“red”, “dia mond”> first, the string “red” is processed discovering that it appears in PC_1 and is associated with lexical item <{<*red*, (1, 2)>}, “red”>. With similar success, “dia mond” is found to be a member of PC_2 with associated lexical item <{<*diamond*, (1, 2)>}, “dia mond”>. The string “dia mond” is further dissected and processed in the same recursive function. Neither “dia” nor “mond” appear in PCs. With results for “red” (appears in PC_1) and “dia mond” (appears in PC_2), the model searches for a rule that can combine members of these categories, discovering rule (16). The rule is instantiated to yield <{<*red*, (1, 2)>, <*diamond*, (1, 2)>}, “red dia mond”>. A possible meaning for the entire string “red dia mond” is, therefore, {<*red*, (1, 2)>, <*diamond*, (1, 2)>}. The comprehension algorithm searches for additional results using the alternative dissection, <“red dia”, “mond”>. No further results are derived. The string “red dia mond” is correctly identified as {<*red*, (1, 2)>, <*diamond*, (1, 2)>}.

In some cases, more than one meaning is derived for a single string. Each string can map to a non-formulaic result, through no use of rules, as well as formulaic results, through the use of rules. Comprehension reintroduces a form of homonymy into the model. “The red cross” can refer to the Red Cross Foundation and “the red square” to the square in Moscow just as likely as their geometrically shaped counterparts employed in this study. As long as multiple meanings provide plausible interpretations for strings, they are useful. String interpretation should reduce the semantic burden in communication, not necessarily produce a single, unambiguous interpretation.

As training data are added to the model, lexical items, rules, and PCs are derived. PCs often include lexical items that express English like PCs, found in (17)-(19). PC membership grows as more training data are added. At times, more than one PC appears to express the same string set membership, but at different stages of development. For example, (17) represents the full set of colors available to the model, while (18) and (19) express subsets of (17).

17. <{“red”, “blue”, “pink”, “green”, “white”, “black”, “ye low”, “gray”, “lime”, “pur ple”}, ((1, 2))>
18. <{“red”, “white”, “black”, “lime”}, ((1, 2))>
19. <{“ye low”, “gray”, “pur ple”}, ((1, 2))>

During comprehension, PCs are substitutable for one another if they appear to express the same string member set, but at different stages of development. (17)-(19) are all considered substitutable for one another. Given the string “white”, PCs (17)-(19) are all representative; (17) and (18) as “white” is a member of their string sets and (19) as it is a subset of (17).

PC substitutions allow abstract categories such as adjectives to form faster. During training, it is common for PCs like (17)-(19) to form. Each of these PCs are created through the derivation of different rules but all appear to suggest the inclusion of an adjective. Abstract categories such as noun, adjective and verb are not necessarily present in young language learners. Studies show that children acquire language in an item-based, piecemeal fashion (Tomasello, 2000). Verb analysis, in particular, shows an uneven usage. For example, a child may only use the word “cut” according to the sentence frame “cut ___”, while “draw” may be used in a variety of manners such as “draw ___”, “draw ___ on ___”, “draw ___ for ___”, and “___ draw on ___”. This suggests that the abstract category of verb is not yet in place, since the verbs are employed with different constraints. This model reflects a similar 'verb island' formation but with adjectives and nouns. PC substitutions allow the islands to be connected.

The model is computationally expensive to implement in both learning and comprehension. Regularities in training data are maximized through a small number of pattern matching mechanisms. Although pruning strategies have been considered, none have been adopted due to lack of success. The approach remains computationally expensive, a serious concern when the target language is scaled-up.

Model Behavior

The model is tested to investigate the emergence of the holophrastic and early multi-word stages. The first correct non-formulaic (non rule-based) and formulaic (rule-based) interpretations signal the beginning of the holophrastic and early multi-word stages respectively. The model is trained with 10 sets of 65 randomly generated <event, description> pairs. Results presented are an average over the 10 sets.

The Developmental Shift

The model is tested for interpretation of 120 strings (10 colors, 10 shapes, and 100 color shape combinations). Each string interpretation yields a set of possible meanings. Correct meanings are charted in Figure 1 depending upon how they are derived (non-formulaically, or formulaically).

For three epochs, there are no successful string interpretations, creating a pre-linguistic period. The first correct interpretation emerges at epoch four and is non-formulaic. This is the model's first word, signaling the onset of the holophrastic stage. Being non-formulaic, the word-to-meaning mapping is representative of first words in child language development. In one set of data, the model's first word is “pen ta gon”, appropriately associated with {*pentagon*, (1, 2)}. For 10 epochs, lexical insights emerge with an increasing volume of correct non-formulaic string interpretations. All strings are representative of single words, either colors or shapes, and never word combinations. At epoch 14, the first non-formulaic word combination is accurately interpreted. This non-formulaic interpretation of a word combination spurs syntactic activity. The first formulaic interpretation is successfully derived at epoch 14, signaling the onset of the early multi-word stage. The emergence of syntax following a period of lexical activity is consistent with child language development.

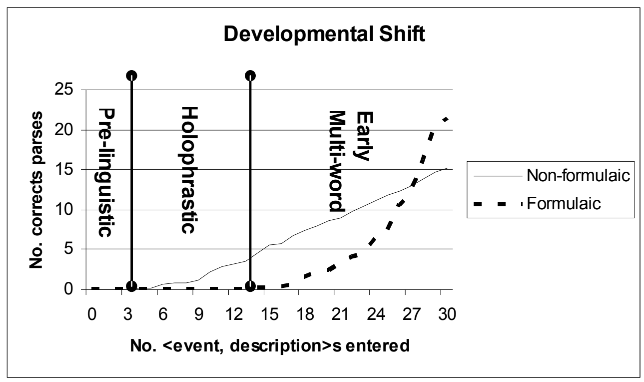

Figure 1: Number of correct non-formulaic and formulaic interpretations.

This result demonstrates two emergent properties in the model; lexical and syntactic awareness. From the outset, the model shows no lexical or syntactic awareness. After a short period of inactivity, lexical awareness emerges, evidenced by the acquisition of first words. The holophrastic stage continues unperturbed for a period before syntactic awareness emerges. Given a larger and more varied training set, that is representative of child linguistic exposure, the periods are predicted to lengthen.

Lexical and Syntactic Expressivity

The model is tested for non-formulaic interpretation of 20 strings (10 colors, 10 shapes), and formulaic interpretation of 100 strings (color shape combinations). Each string interpretation yields a set of possible meanings. Correct meanings are charted in Figure 2 depending upon how they are derived (non-formulaically, or formulaically).

The distinction between non-formulaic and formulaic language is clear. The former makes no use of rules while the latter does make use of rules. Formulaic language is most expressive when rules are applicable to large sets of data i.e. phrasal category string membership is high. This model identifies a formulaic relationship at epoch 14. The relationship is representative of the English grammar rule NP = Adj. N. On establishing this formulaic expression, the PCs representing adjectives and nouns, constrain rule expressivity. A correlation between the percentage of lexical items acquired and the expressivity of the formulaic expression exists. PC membership swells as subset and superset relationships are derived, allowing abstract categories to form.

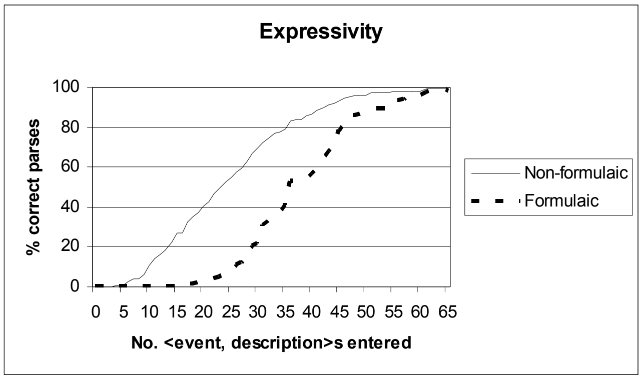

Figure 2: Percentage of correct formulaic and non-formulaic interpretations.

This result demonstrates that the expressive power of syntactic rules is correlated with the number of lexical items correctly identified in the model. As lexical membership increases, PC string membership expands, and rules become more expressive. This finding is consistent with child language acquisition. As phrasal categories form, they become increasingly abstract and employed by a number of rules. Given more strict PC connectivity constraints, Tomasello's (2000) verb island effect is predicted.

Conclusion

The model demonstrates two behavioral shifts that are present in child language development. First, lexical awareness emerges as syllable combinations are recognized as expressions of word-to-meaning mappings. This period persists in the absence of syntactic awareness. Second, word combinations are recognized as expressions of syntactic relationships. Syntax emerges and becomes increasingly expressive as training continues. The item-based acquisition strategy can acquire language in a child-like manner through exploiting a small number of cognitively general learning mechanisms.

Acknowledgments

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

References

Bates, E., & Goodman, J. C. (1999). On the emergence of grammar from the lexicon. In B. MacWhinney (Ed.), The emergence of language.

Brent, M. R., & Siskind, J. M. (2001). The role of exposure to isolated words in early vocabulary development. Cognition, 81, 33-44.

Dehaene-Lambertz, G., & Houston, D. (1998). Faster orientation latency toward native language in two-month-old infants. Language and Speech, 41, 21-43.

Dominey, P. F., & Boucher, J.-D. (2005). Developmental stages of perception and language acquisition in a perceptually grounded robot. Cognitive Systems Research, 6(3), 243-259.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. Cognition, 48 (1), 71-99.

Feldman, J. A., Lakoff, G., Stolcke, A., & Weber, S. H. (1990). Miniature language acquisition: A touchstone for cognitive science. Proc. 12th Ann. Conf. Of CogSci Soc.

Houston, D. M., Santelmann, L. M., & Jusczyk, P. W. (2004). English-learning infants' segmentation of trisyllabic words from fluent speech. Language and Cognitive Processes, 19(1), 97-136.

Jack, K., Reed, C., & Waller, A. (2004). A computational model of emergent simple syntax: Supporting the natural transition from the one-word stage to the two-word stage. Coling, 20th Conf on Comp. Ling., Geneva, Switzerland.

Kellman, P. J., Gleitman, H., & Spelke, E. S. (1987). Object and observer motion in the perception of objects by infants. Journal of Experimental Psychology - Human Perception and Performance, 13(4), 586-593.

Kirby, S. (2001). 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.

MacWhinney, B., & Bates. (1989). The crosslinguisitic study of sentence processing. New York: Cambridge University Press.

Markman, E. M., & Wachtel, G. F. (1988). Children's use of mutual exclusivity to constrain the meanings of words. Cognitive Psychology, 20(2), 121-157.

Mehler, J., Dupoux, T., Nazzi, T., & Dehaene-Lambertz, G. (1996). Coping with linguistic diversity: The infant's viewpoint. In J. L. Morgan & K. Demuth (Eds.), Signal to syntax. Mahwah, N.J.: Lawrence Erlbaum.

Quinn, P. C. (2003). Concepts are not just for objects: Categorization of spatial relation information by infants. In D. H. Rakison & L. M. Oakes (Eds.), Early category and concept development: Oxford University Press.

Regier, T. (2005). The emergence of words: Attentional learning in form and meaning. Cognitive Science, 29(5).

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. Cognition, 61(1-2), 1-38.

Steels, L. (2001). Language games for autonomous robots. IEEE Intelligent systems, 16-22.

Tomasello, M. (2000). The item-based nature of children's early syntactic development. Trends in Cognitive Sciences, 4(4), 156-163.

Wray, A. (2002). Formulaic language and the lexicon. Cambridge: Cambridge University Press.