A Parallel Approach to Modeling Language Learning and Understanding in Young Children

Charles Hannon and Diane J. Cook

University of Texas at Arlington
Box 19015
Arlington, TX 75019
{hannon,cook}@cse.uta.edu

Abstract

To reduce the complexity of studying a parallel mechanism for natural language learning and understanding which supports both utterance and discourse processing, we propose a computational model demonstrating the performance of a young language user just starting to construct two and three word utterances. Our model attempts to capture the explanatory aspects of parallel language processing and the division between long and short term memory which have been shown as important factors in human language learning and performance. We also address the pragmatic consideration of building and supporting the resulting parallel system on a cluster of SMP computers.

Introduction

Attempts to model the complete process of language learning and understanding of adult speech have proven overwhelming in their complexity. While the ultimate goal of natural language research is a flexible and extensible model of the complete process of adult speech use and learning, most research efforts have been forced to restrict their domain to something more quantifiable. If an explanatory model for language processing is used, the problems's complexity can be reduced by first addressing a learning and understanding process which demonstrates the capacities of a young child just learning to form two and three word sentences from her acquired lexical entries. This phase of language development is often referred to as telegraphic speech (Fromkin & Rodman 1984).

Our computational model attempts to capture the process of language understanding and learning during the telegraphic speech stage of development. By capturing this language processing aspect of cognition, we hope to build a generalized model of language understanding which can be incorporated into systems such as robots, intelligent search tools and data miners. The direct language processing elements are supported by a simplified set of vision elements, allowing the study of visual symbol grounding. They are also supported by a set of Higher Order Process (HOP) elements used to drive the overall intentionality of the system and process the higher level language functions of discourse, conversation and social acts (Austin 1962, Searle 1969)

The model is based on a multi-agent architecture where each element within an agent is a multithreaded process. All communication between elements is message-passing based relying on our own Adaptive Modeling environment for Explanatory Based Agent (AMEBA) parallel environment. AMEBA runs on either a standard set of distributed computers or a cluster of high-speed networked SMP machines running Linux. The environment is based on concepts from both Beowulf and PVM (Merkey, 1999).

Theoretical Foundations

The telegraphic speech stage normally occurs at about 24 months of age after the child has been using holophrastic (or single word) sentences for over six months. It is characterized by the absence of function words, tense and word agreement. Thus, the telegraphic speaker demonstrates only the simplest syntactic relationships between words in generated utterances while still being able to perform a semantically rich set of useful speech acts. While being limited by the amount of stored syntactic and semantic information, the child at the telegraphic speech stage is still capable of understanding a great deal of utterance and discourse performed at a normal adult level of speech.

Being based on Principles and Parameter theory, and thus, an assumption of the existence of the Language Acquisition Device (LAD) and a Universal Grammar (UG), our research views the process of language learning as both the setting of UG settings and the collection and storage of relevant language data. We view the language generation, understanding, and learning tasks as occurring within the context of parallel lexical, syntactic and semantic processing that is embedded within the overall conceptual reasoning and control of the unified cognition. Within the collection of the utterance processing elements, we try to model the telegraphic speech user's ability to rely on both adult and other child language uses to reinforce both the speech act and general language performance both directly and indirectly.

To study unified utterance and discourse understanding and learning, our language model relies in both Jackendoff’s model of conceptual reasoning and the discourse theories of Speech and Conversation Acts. To study non-language symbol grounding, it supports simple vision and concept processing. The internal model of these processes is very pragmatic; however we avoid obvious non-explanatory reasoning methods by grounding these on cognitive physiology research.

AMEBA Architecture

The AMEBA architecture represents the refinement of generalized parallel tools to produce an environment for testing cognitive theories. The architecture runs on a SMP cluster using a Beowulf-like multiple high-speed networks. It relies on a process control daemon running on each SMP node to manage processes for a centralized set of X-window user interfaces.

AMEBA provides an explanatory computational architecture on which explanatory cognitive models can be constructed and tested. In its computational architecture, it attempts to capture the explanatory force of a connectionist neural model while allowing the use of the better understood representation and reasoning methods of symbolic AI. From a system perspective, it provides processor transparency within a parallel system and a flexible method of process and knowledge management. The key element in the solution of both of these sets of requirements is the etheron process template shown in Figure 1. An etheron provides a container for an instance of any inference or routing mechanism needed by the system. Once contained, the etheron supports the mechanism with, 1) a standard way to load and store knowledge, 2) interfaces to AMEBA's management tools and 3) a generalized set of communication channels for talking with other etherons.

Etherons can be connected to each other in a lattice, like the neurons in a neural network, but we have found it easier to model agents using a tree structure. While they are somewhat similar in design to an artificial neuron, etherons function more like a neural analyzer, a sub-network in the brain which serves a particular processing function (Martindale 1991). Intra-agent communication between etherons is defined using a set of stimulus messages. These messages are defined in such a way as to reasonably simulate the level of information being passed between real neural analyzers within a collection of neural stimuli. The inter-agent communication process is also based on message passing, but the architecture leaves the details of the message content up to the system designer.

Using AMEBA's management software, a user can dynamically build a system of agents out of a set of predefined etheron types and control the internal knowledge loaded into each etheron during operation. Since etherons support the ability to be started, stopped and moved independently, the user can dynamically select which portion of a parallel system to activate. Etherons are internally multithreaded to take full advantage of a SMP host processor. During the building of an etheron type, the builder can use predefined inference methods or build a special method for the particular task at hand. The AMEBA libraries provide support for a temporal logic based production system, a temporal-modal semantic network, a back-propagation ANN, and a generic database lookup tool built on top of PostgreSQL, an open-source Object-Relational DBMS.

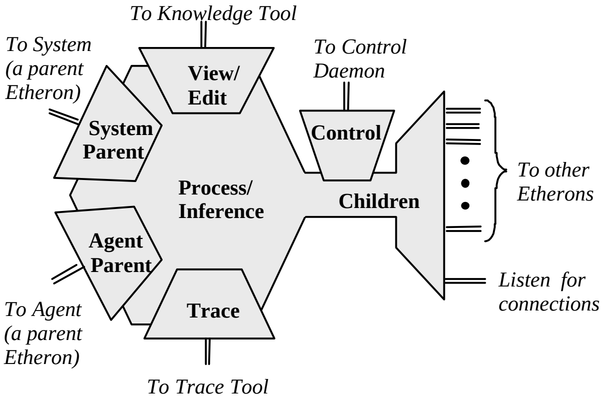

Figure 1. The Etheron

The Tallus Test System Design

Using AMEBA, we have constructed a Teacher Assisted Language Learning and Understanding System (TALLUS) consisting of a teacher agent and two student agents. The current teacher agent only provides a text-based interface for entering adult level utterances and capturing the telegraphic speech level utterances of the students. The architecture of one of the student agents is presented in Figure 2. Each student agent is constructed from 20 etherons of six different types. Since a student agent must glean enough information out of an adult speech utterance.to learn and use telegraphic speech, the majority of the language processing is dedicated to understanding. One reason for picking the telegraphic speech level to study is that the agent has to do very little to the underlying semantic structure of an utterance to speak it. Communication between agents TALLUS system uses broadcast messages to simulate the language learning during immersion.

TALLUS divides knowledge storage in the agent into Long Term Memory (LTM) and Short Term Memory (STM) storage (Baddeley 1987, Levine 1991). While being strongly influenced by the Baddely & Hitch's

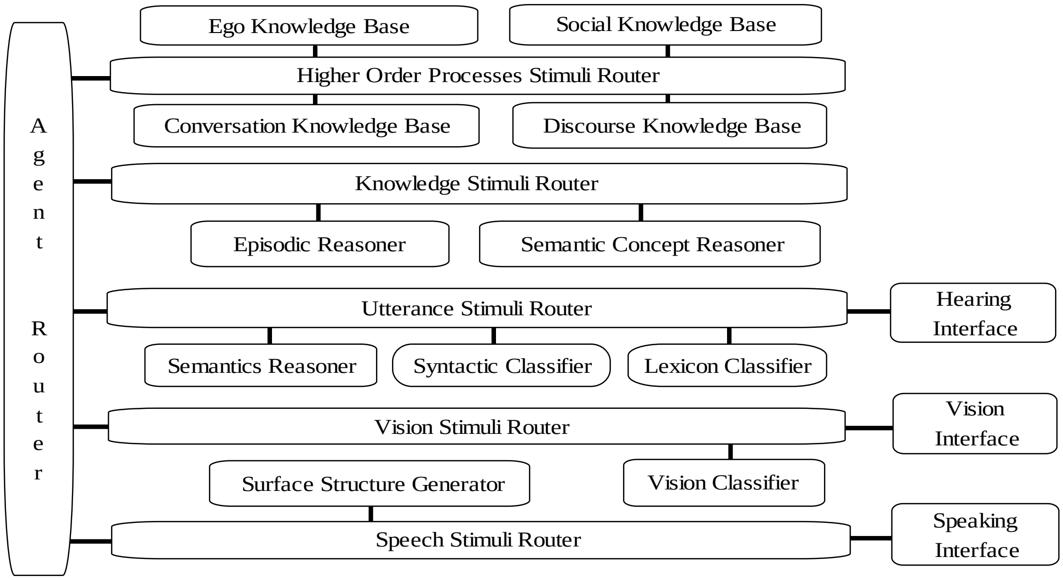

Figure 2. A Student Agent in TALLUS

working memory model, the concept of STM is extended past their phonological loop and visuo-spatial scratch-pad to provide a seperate STM element in all three classifiers and the Semantic Reasoner. We have been able to use a set of three reasoners (a production system, semantic network, and database based pattern matcher) for the Long Term Memory (LTM) storage in all etherons. The Short Term Memory (STM) elements of each etheron below the HOP processes has been specialized.

The Vision Classifier is a simple simulation containing a STM representation which functions much like a conceptualized visuo-spatial scratch-pad. The Lexical Classifier uses a STM that closely reassembles a phonological loop.

The Syntactic Classifier uses a stored set of flat (non-tree) part-of-speech patterns to generate a governor and government relationship for most non-function words in an utterance. While it is based on Principle and Parameters theory and can extract most information needed for telegraphic speech, research has demonstrated the need to move toward a more formal X-bar syntax. The Semantic Reasoner uses the AMEBA's generic semantic network for its LTM storage. The STM portion of this etheron creates and maintains an activation list that supports such explanatory concepts as priming and recall. All of the HOP etherons use the AMEBA's generic production system for both LTM (rules) and STM (facts) storage.

Information is passed between etherons in an agent using stimuli messages. Since the goal of an etheron network is to emulate a collection of neural analyzers, stimuli messages have been kept as simple as possible. The basic structure of a stimulus is:

[x]STN(parameter[0], parameter[1], ... parameter[n])

where x is an optional modifier and STN is a three letter stimuli type name. AMEBA's communication API takes care of encoding and decoding the parameter list but enforces no semantic structure on the list,leaving this to the sending and receiving etheron. For example, in TALLUS, a TAG stimuli type is used to tag an input word with its speaker and the order in which it is received, and a POS stimuli type is used to assign a part-of-speech to a tagged word. All etherons in the language sub-network know how to extract the information encoded in both of these stimuli types to do their particular task.

By using the modifier field of a stimuli type, additional semantics are added to a stimuli type's meaning. For example, an unmodified POS stimuli is always interpreted as a positive assertion about the part-of-speech of a word. If an etheron cannot absolutely assert the part-of-speech of a tagged word, but has evidence that it might be some part-of-speech, it can send a *POS stimulus which is interpreted as a positive hypothesis about the contents of the POS stimulus. If it has evidence that a certain tagged word cannot be some part-of-speech, it can send a !POS stimulus which is interpreted as a negative hypothesis about the contents of the POS stimulus. A ?POS stimulus is used to ask a question about the truth of contents of a POS stimulus.

As words arrive at the Agent Interface, they are converted to TAG stimuli and these stimuli are broadcast to all etherons directly connected to the Utterance Stimuli Router (the green colored etherons in Figure 2). After the TAG message is sent, the resulting processing becomes highly parallel. While the Lexical Classifier is trying to create a POS stimulus for the word, the Syntactic Classifier is trying to construct sentences out of the input words and relate these words to the part-of-speech information retrieved by the Lexical Classifier. Even before the POS message gets to the Syntactic Classifier so it can generate a SYN (syntax) stimulus to send to the Semantic Reasoner, the Semantic Reasoner is activating the word nodes in its LTM and starting to generate the SEM (semantic) stimuli it will broadcast to the HOPs and vision system based on the total utterance. At the HOPs, these SEM stimuli are being matched with each other and the REF (reference) stimuli being generated by the vision sub-network to fire rules of discourse, socialization and planning which determine both the response needed and how this response should be carried out.

Learning in our system is incorporated as part of each etheron's normal knowledge processing, and not as any separate or distinct machine learning algorithm. Utterance level learning occurs when a higher level system is able to feed-back a hypothesis it used in place of the data it would normally receive from an input level system. For example, both the Syntactic Classifier and Semantic Reasoner use part-of-speech information to complete their task. If the Lexical Classifier is unable to send a POS stimulus, the Syntactic Classifier and Semantic Reasoner try to figure out the part-of-speech of a word by context. These hypotheses are then fed back to the Lexical Classifier which either adds a new part-of-speech record to its LTM or modifies its confidence in a record. Records that fall below a certain confidence level are removed from the LTM all together. AMEBA also supports the dynamic modifiication of a production system's rules (i.e., live experts) which we will be using in the future for discourse level learning.

Results

Currently TALLUS demonstrates, 1) an understanding of telegraphic language at both the utterance and discourse level, 2) the ability to associate an input utterance to its visual environment and 3) the learning of concepts and language at the utterance level. TALLUS operates on a SMP cluster of 16 total processors and two redundant single processor servers. The topology of this system is eight two-processor SMP machines connected via two switched and one hubbed 100baseT networks.

While improving speedup performance is not the major focus of our current research, AMEBA relies on the ability to distribute large cognitive models across a number of computers. Figure 3 presents the results of a set of speedup tests conducted on 1 to 5 two-processor SMP nodes. The system was first divided across agent boundaries, and then, each Student agent was divided across two SMP machines. The tests were connected using two network topologies, one with 100MB/sec and one with 500MB/sec effective throughput.

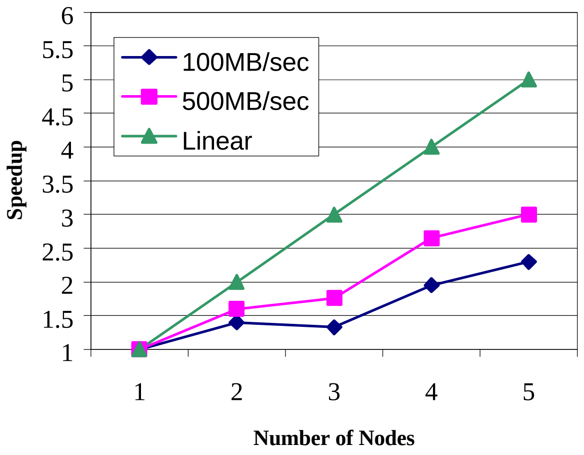

Figure 3. TALLUS Performance

The reported results represent the average trial time of the values that clustered around the median time of test. This method was used instead of a straight average since some evaluation tests took as much as three times as long as the median. While access to the high-speed networks being used to link our SMP cluster is carefully controlled, it is suspected that some of these longer test times were a result of network interference caused by utility messages within the network. While not providing anything close to linear speedup, these results show that AMEBA’s speedup is well within the expected range of such a course grain parallelization method.

As a brief example of the system's language understanding performance, take the simple teacher input utterance; "What is this?" [spoken with the scene database containing the representation, item0 (the teacher) contains item4 (a blue ball)]. To further simplify, we will assume that all utterance level etherons already have the knowledge to process this sentence so that no learning is required.

Given this text input, the Agent Interface receives and tags the words and the Lexicon Classifier creates three part-of-speech stimuli for them. The Syntax Classifier uses this information to look up the phrase template and generates two syntax stimuli that state; 'in sentence 1, word 1 is the subject and word 3 is the object of word 2'. Using all of this, the Semantic Reasoner first activates all possible meanings of the three words, and then, uses the syntactic information to reduce the 'is' to the equative (or linking) form and the 'this' to a grounding marker. It then generates two semantic stimuli that state; 'in sentence 1, the teacher requests the name of ground marker1 and the teacher contains ground marker1'.

The semantic stimuli cause the Ego HOP to generate a stimuli stating 'for sentence 1, I want to answer'. The semantic stimuli causes the social HOP to generate a stimuli stating 'for sentence 1, I should answer' since the teacher has not told them to be quiet. The Conversation HOP generates a stimuli indicating the agent has the turn. Using all of this, the Discourse HOP generates a reference question stimuli asking; 'regarding sentence 1, how can the statement, teacher contains ground marker1, be resolved'. The Vision Classifier then looks in the scene to see that the teacher contains item4. If it has a correct symbol name for item4, it returns two reference stimuli that state 'regarding sentence 1, the teacher contains ball and blue is the color of ball'. The Discourse HOP then generates three action stimuli that state; 'activate teacher, activate ball in reference to teacher, and activate blue in reference to ball'. These action stimuli are used by the Semantic Reasoner to generate two response stimuli that are sent to Agent Interface. Using these the Agent Interface speaks; "blue ball". If the Vision Classifier has no reference for item4, no output is generated but the HOPs start listening for the other student or teacher to give the answer. Once the answer is confirmed by the teacher, the student updates its knowledge (i.e. learns) that item4 is a blue ball.

As an example of the system's learning performance, take a set of input utterances containing the word ‘ball’ where the agent starts with no information about the word. When a sentence with ‘ball’ is received, the Lexicon Classifier creates part-of-speech stimuli for the known words. The Syntax Classifier looks up all patterns that match the phrase length and known part-of-speech slots and generates a positive hypothesis about the missing part-of-speech. This positive hypothesis would then cause the Lexicon Classifier to begin creating a belief that `ball' is a noun.

Given enough examples of `ball' filling a noun position, the Lexicon Classifier's belief grows to the point that it will assert that `ball' is a noun when stimulated by a tag stimulus. Now that the part-of-speech of `ball' is known, the Semantic Reasoner will try to activate the semantic concept `ball-noun' from an input sentence but will fail because it does not exist. This causes the Semantic Reasoner to create a new node for `ball-noun' and link it, with a minimum level of belief, to the concept `physical-object' since it, like `physical-object', is capable of having a color. As the concept `ball-noun' is used in other contexts, additional semantic and conceptual links are formed.

Learning new phrase patterns and syntactic roles is similar to the word learning process but is too complex to describe here. The current system uses a non-tree approach that has proven to not scale. It is being replaced with a method which relies directly on X-bar syntax.

Future Work

Clearly, the current work with TALLUS can be extended to gain a further understanding of both the language processing and its impact on a broader unified model of cognition. AMEBA is designed to allow cognitive models to be further and further refined. Our current ability to model discourse processing, general vision, and higher order processes needs to be both broadened and refined. While TALLUS is the major focus of our research with AMEBA, we are also working on other more detailed models of more well defined cognitive tasks. As expected, these systems have already pointed out several minor weaknesses in TALLUS which need to be addressed. It is this interplay of models which we hope will allow AMEBA to contribute to the overall field of cognitive science.

The next detailed modeling domain we plan to address is the Wisconsin Card Sorting Test (WCST). The WCST will provide an opportunity to study such things as mental sets and task planning. The results of this model will then be feedback into TALLUS.

Conclusion

TALLUS has supplied both insight into parallel language processing and the possibilities of unified model of cognition. It is probably best for cognitive scientists to view unified cognitive modeling like physicists view controlled nuclear fusion. It may not be something we will see in our lifetime, but it will not invent itself. Using AMEBA to generate models like TALLUS, and then, combining the results of these models into an unified whole may be one way to approach such a formidable task.

TALLUS shows that it is possible to build a cognitive model of language learning and understanding that can be examined symbolically. As both AMEBA and TALLUS mature, TALLUS should both continue to improve our understanding of the cognitive processes of language learning and understanding and serve as model of applications applying these understandings.

References

Anderson, J. R. 1995. *Cognitive Psychology and its Implications*. New York: W. H. Freeman and Company.

Austin J. L. 1962. *How to do things with words*. Boston: Harvard University Press.

Baddeley, A. D. 1987. *Working Memory*. London: Oxford University Press.

Fromkin, V. and R. Rodman 1988. *An Introduction to Language*. Fort Worth, TX: Holt, Rinehart and Winston, Inc.

Haegeman, L. 1991. *Introduction to Government and Binding Theory*. Oxford, UK: Blackwell.

Martindale, C. 1991. *Cognitive Psychology, A Neural-Network Approach*. Brooks/Cole Publishing Co., Pacific Grove, CA.

Levine, D. S. 1991. *Introduction to Neural and Cognitive Modeling*. Hillsdale, NJ: LEA, Inc,.

Searle, J. R. 1969. *Speech Acts: an essay in the philosophy of language*. London: Oxford University Press.