Learning Language in the Service of a Task

Mark F. St. John
Department of Cognitive Science
University of California, San Diego
La Jolla, CA 92093-0515
stjohn@cogsci.ucsd.edu

Abstract

For language comprehension, using an easily specified task instead of a linguistic theoretic structure as the target of training and comprehension ameliorates several problems, and using constraint satisfaction as a processing mechanism ameliorates several more: namely, 1) stipulating an *a priori* linguistic representation as a target is no longer necessary, 2) meaning is grounding in the task, 3) constraints from lexical, syntactic, and task-oriented information is easily learned and combined in terms of constraints, and 4) the dramatically informal, "noisy" grammar of natural speech is easily handled. The task used here is a simple jigsaw puzzle wherein one subject tells another where to place the puzzle blocks. In this paper, only the task of understanding to which block each command refers is considered. Accordingly, the inputs to a recurrent PDP model are the consecutive words of a command presented in turn and the set of blocks yet to be placed on the puzzle. The output is the particular block referred to by the command. In a first simulation, the model is trained on an artificial corpus that captures important characteristics of subjects' language. In a second simulation, the model is trained on the actual language produced by 42 subjects. The model learns the artificial corpus entirely, and the natural corpus fairly well. The benefits of embedding comprehension in a communicative task and the benefits of constraints satisfaction are discussed.

Introduction

Understanding language, and particularly learning to understand language, is a tricky task. A variety of imposing practical and theoretical problems stand in the way. This paper addresses four of these obstacles and shows ways to surmount them and grasp a better understanding of language learning and understanding along the way.

The four obstacles are 1) how to specify a satisfactory representation of the meaning of the language input, 2) how to ground the semantics of concepts used in the communication 3) how to combine lexical, syntactic, and task-oriented information to produce comprehension, and 4) how to handle incomplete and grammatically "noisy" language such as natural spoken language.

The message of this paper is that using constraint satisfaction as a processing mechanism and employing the fact that language is learned and used in the service of performing a task goes a long way toward surmounting these obstacles. First, I'll briefly discuss each obstacle in turn and show how these two ideas can address them. Then I'll present two simulations that show these two ideas at work. This paper concentrates on the first two obstacles and points to preliminary work to address the remainder.

Obstacles

The first obstacle is the need to stipulate a linguistic-theoretic representation as the target or result of comprehension. One well known and useful representation is thematic case roles, such as agent and patient (Fillmore 1968). Unfortunately, case roles are known to either proliferate in number or to become inexact as situations become more diverse and complicated. Specifying the features that characterize concepts is similarly difficult.

Additionally, all but the most trivial language requires something like embedded propositions to specify the relations between concepts. Such symbolic representations impose strong assumptions about the representation of the results of comprehension and place a particularly heavy burden on Parallel Distributed Processing (PDP) models since propositions are virtually impossible to represent in a single vector of units.

Several PDP models have attempted to handle this problem. Miikkulainen and Dyer (1991), St. John (1992), and Touretzky and Hinton (1985) represented multiple propositions concurrently in a hidden layer. Individual propositions could be pulled out one at a time for

inspection. Not only is this pull out scheme awkward, but individual propositions still cannot contain embeddings since there is still no way to represent the case where one argument in a proposition is a whole other proposition. Pollack's RAAM model (1988) allowed distributed representations in the output layer, but training still required these representations to be unpacked into their fundamental components.

The solution proposed here is to define the target to be some easily represented task. For example, imagine two people solving a jigsaw puzzle where one person commands the other where to place the blocks. The listener's job is to understand the language input and then move a block. The target can be a simple representation of the updated state of the puzzle after each command. The benefit is that the task is easy to represent, and the complex linguistic structure of the language is hidden inside the listener. If we make the listener a PDP model, the output can be the task, and any linguistic structures required to compute the output reside internally in the hidden layers.

The second obstacle is the need to define the meaning of concepts in the language. Again, the researcher may be required to specify this information, for example, by coding a set of semantic features for each concept. In contrast, Allen (1987), Elman (1990), Miikkulainen and Dyer (1991), and St. John and McClelland (1990) showed that the task itself can be used to specify the semantics. That is, the needed semantics can be learned by a model in the service of solving some task. These models learned which concepts are seen with which other concepts. Semantics, in these models, is defined by the co-occurrence statistics of the concepts in the corpus of training examples.

The approach taken in the puzzle task is to have the model learn the meanings of words like "big" and "blue" and the ramifications of syntactic forms in terms of their ability to help determine to which block the speaker is referring. Thus, words and syntax are learned to be defined in terms of their communicative functions. This idea of language meaning as language use is developed much more fully by Clark (1985). Having the target actually be some performed task, as is the case in this paper, makes this point especially clear.

The third obstacle is the need to combine information from words, syntax, and the task to understand the command specified by the speaker. Speakers often rely on the situation to convey important information. Constraint satisfaction is a powerful mechanism for performing this process. Each piece of information from any source is viewed as a separate constraint. These constraints are combined to compute a single, coherent interpretation (cf. St. John, 1992).

The fourth obstacle is the need to handle the grammatical informality of natural speech. Informal constructions such as repetitions, restarts, and ellipses are so common that they really are the rule and not the exception. As such, they must be treated within the normal course of processing. Constraint satisfaction is well suited to this task. When the language is viewed as a set of constraints, the important factor is that there be sufficient constraints to specify the communication, not that the communication correspond to a specific grammatical form.

Recent models in the literature that handle informal grammatical forms (Lehman, 1990) and ellipsis (Frederking 1988) first assume that input language will be grammatical, and only when the normal processes fail do they perform a time-consuming search for corrections or additions that will produce grammatical and sensible parses. The constraint satisfaction alternative is to use the available constraints to compute an interpretation and only concern itself with the grammar to the extent that it informs that computation.

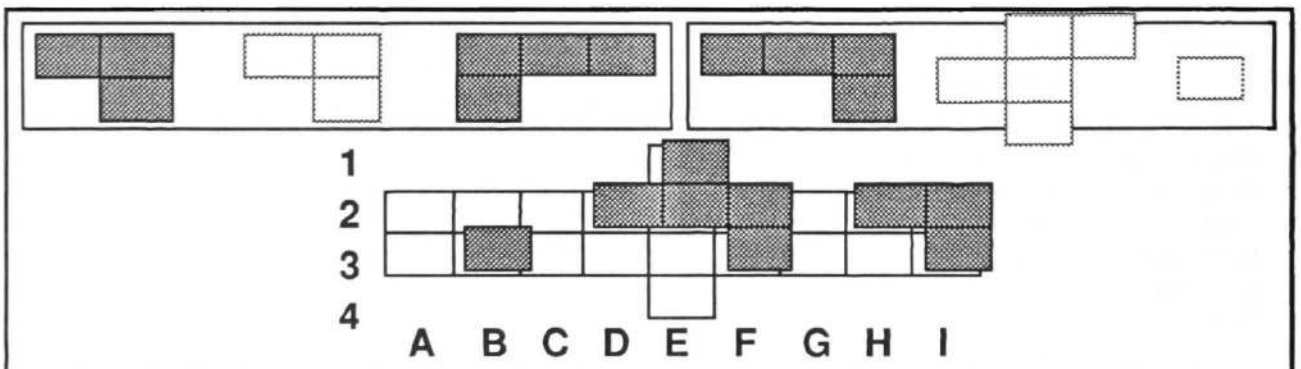

Figure 1. The jigsaw puzzle. Pieces of varying shapes, sizes, and colors are place on an outlined figure.

Task

The task to be use here is the simple puzzle task introduced above. Subjects, UCSD undergraduates, were asked to solve a series of three simple jigsaw puzzles. Each puzzle consisted of six wooden blocks of varying shapes, sizes, and colors that had to be placed on an outlined figure. Unplaced blocks were arranged on two mats above the puzzle figure (see figure 1). The trick was that the subjects were required to sit on their hands and tell the experimenter where to move the blocks. The figure was labeled with Cartesian coordinates to facilitate this task. Subjects' spoken commands were recorded and transcribed.

Since there is considerable linguistic complexity in just referring to which block was to be moved, I will concentrate on just this block reference task in this paper.

Simulation 1 - Artificial Corpus

Rather than proceed directly to the natural corpus of commands produced by subjects, I will begin with a corpus constructed by hand. An artificial corpus is a good place to start the investigation because features of the language are easily controlled. As a further simplification, this corpus contains only first moves in the puzzle task rather than whole series of consecutive moves. This restriction will be removed in the second simulation.

Commands were composed from a small set of adjectives (small, big, square, red, blue, and green), and phrase types, such as simple noun phrases, relative clauses, and prepositional phrases. The commands also made use of "left", "right", and "middle" that require paying attention to word order, for example, "the left block," "the right block that is on the left page," and "the left block on the left page." Finally, blocks could be referenced in terms of other blocks, for example, "the block on the right of the big blue block."

All together, there were 138 commands. Several provisions were made to insure that the corpus was combinatoric in the sense that roughly equal numbers of commands referring to each of the six blocks, and adjectives and relative terms applied to each relevant block in roughly equal numbers. A combinatoric corpus insures that the model will learn the pure meaning of each term without becoming muddied by biases in frequency. Of course the real world may not be so kind, and the second simulation addresses this issue.

Example Commands

the big blue block that is on the right of the left page
the big block that is on the right of the left page
the big red block that is on the left of the right page
the small blue block
the small red block that is on the left of the big blue block

Architecture. The architecture is a simple recurrent network (Elman 1990) wherein the activation in the internal hidden layer is copied back to the input layer on each consecutive step. A recurrent network is useful because it allows each word to be processed sequentially by the same set of weights, yet allows information from previous words to be carried forward. The model cycles through a command one word at a time, with activations from the hidden layer being copied back to the input on each consecutive step (see figure 2). The input is a command, a representation of the blocks yet to be placed on the puzzle, and the activations from the recurrent hidden layer. One unit in the input layer was used to represent each possible word, and as the model worked through a command, the current word was activated in the input layer and the previous word was removed. While the input layer is therefore local in its representation of individual words, the internal hidden layer is free to develop more efficient distributed representations.

The representation of to-be-placed blocks was simply to activate one unit for each remaining block. There was no feature description of any block. Because this simulation deals only with the first command in the puzzle, all six blocks were activated for each command.

The output and target for the task was to activate one unit for the block referred to by the command. The target is specified after each word.

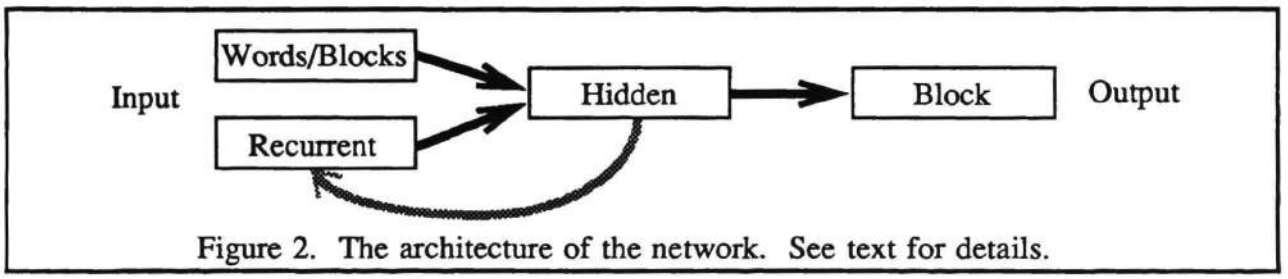

Figure 2. The architecture of the network. See text for details.

That is, the model is asked to produce the correct target even after just the first word of a command is presented. Error between the target and the model's response is used to change the weights of the network via the backpropagation algorithm (Rumelhart, Hinton, and Williams, 1986). This training procedure requires the model to extract the most information from each incoming word.

There were 17 units for words and 6 units for blocks in the input layer, 40 units in both the recurrent and hidden layers, and 6 units for blocks in the output layer.

Results and Discussion. The model was trained for 2000 epochs (trips through the training corpus) with a learning rate of .01. The model mastered the corpus entirely by activating only the correct block for each command in the corpus.

This mastery demonstrates a number of points. First, the model is able to learn and correctly combine a number of partial constraints to activate the correct block. For example, both "big" and "red" refer to two blocks, but together they specify only one block. Each adjective acts as a constraint on the specification of a block. The process of constraint satisfaction embodied in the network works well to combine these constraints.

Second, the model learns to process relative clauses, prepositional phrases, and word order correctly. Commands like "the small block that is on the left of the big blue block" describe two blocks, yet the model picks the correct one as the referent. Commands like "the block that is on the left of the right page" and "the block that is on the right of the left page" refer to different blocks and so require handling the order of left and right correctly.

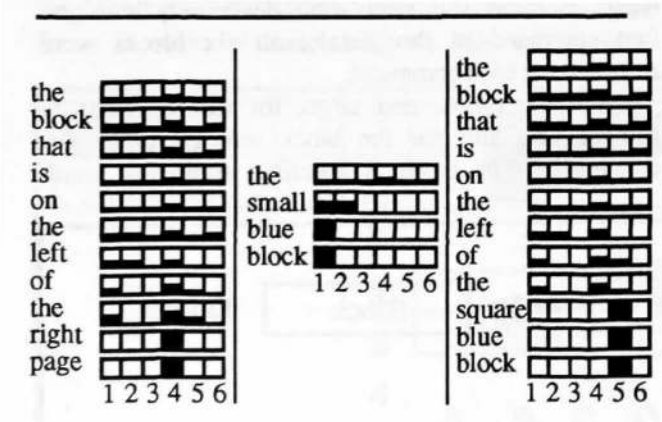

Figure 3. Processing commands. The output activations (blocks 1-6) are shown after processing each word.

A simple way to observe what the network has learned, and to confirm that it has not simply memorized the training set, is to observe the activation of the output units as a command is processed. If the activations throughout the command reflect the constraints on meaning imposed by each new word, we can have greater confidence that the model has learned those constraints. Figure 3 provides three examples. The output activations for the six blocks are shown in black after each word of a command is processed.

Third, the model's understanding of words derives from the task - the words function to differentiate the blocks and determine to which one a command refers. The semantics of the terms, then, is grounded in their functions in the puzzle task.

Fourth, neither the semantics nor any linguistic-theoretic structures had to be specified in the input or target. The input was the sequence of words, and the target was simply the block referenced by the command. To whatever extent case roles, embedded propositions, or tree structures needed to be computed for the task, they were represented and computed internally in the hidden layer of the network.

Simulation 2 - Natural Corpus

A fresh network was trained on a large corpus of commands produced by actual subjects solving the puzzle task. The results from this simulation are preliminary, but they demonstrate important points.

A natural corpus is interesting in a number of ways. Foremost, the grammar used by actual subjects is highly informal and the language is frequently vague since subjects can rely to a large extent on the puzzle situation to convey information. Nearly every command contained repetitions of words or phrases, restarts, or ellipses.

A second interesting aspect of the natural corpus is that subjects solved entire puzzles, and these sequences of block commands demonstrated important task constraints. Most importantly, there was a rough standard order for placing blocks on the puzzles: essentially from the most constraining block to the least constraining block. There was substantial variability in the ordering, but statistically, an ordering is evident. If the comprehension system can utilize these constraints, it can resolve otherwise ambiguous language and it can generally ease its comprehension task when these constraints are redundant with the language of a command. For example, if one of the two red blocks has already been placed, the otherwise ambiguous reference "the red block" becomes clear.

Finally, subjects were allowed to change the positions of blocks and start over. Therefore, consecutive references to the same blocks could occur. Subjects almost invariably used relative terms like "it" or "those blocks." Another relative term is "the block on the left." After the first block on the left is placed, the second block becomes the new block on the left. The model must bind these relative terms to the correct block on each occasion.

Architecture. The architecture was identical to the previous architecture except that many more words were used by subjects, 613, and therefore more words were represented in the input layer. The one difference in training was that the representation of to-be-placed blocks was updated from command to command throughout the course of solving a puzzle. Between commands, the activation of the hidden layer was copied to the input layer, as within commands. The recurrent layer was only reset to zero activations between puzzles. The corpus contained 42 speakers and a total of 1495 commands.

Results and Discussion. The model was trained for 400 epochs with a learning rate that gradually dropped from .005 to .002. The model activated the correct block most strongly on 78% of the commands. The trained model was also tested on a corpus of 3 novel subjects. It performed correctly on 53% of the 134 novel commands. These figures are certainly not great, but an examination of the model's successes and failures is illuminating.

First, the model handles the informality of the grammar very well. Ellipsis of nouns and entire phrases, restarts, and repetitions cause no trouble for the model. For example, commands like "the blue, the big blue, no, yeah, the big" are processed correctly.

The model also handles the relative terms "it" and "left" correctly, and uses the task constraints to understand otherwise ambiguous references like "the red block" discussed above.

The model acquires the rough standard order of block placements readily, and even too well. That is, on many occasions, a subject will violate the standard order. Depending on the violation, the model will either follow the language or follow the standard order. More egregious violations of order, for example choosing the smallest, least constraining block first, are quite rare in the corpus. In these cases, the model will override the language input and activate the standard first block.

To some extent this effect is reasonable, though overly strong. A number of researchers have found that semantic constraints can override the language input. A telling example is to ask subjects "how many animals of each kind did Moses take on the ark?" Most subjects respond "two" without noticing that Moses is not the correct biblical figure (Erickson and Mattson, 1981; and Reder and Cleeremans, 1990). This effect can be viewed in terms of constraint satisfaction as a case where conflicting constraints are present. The task constraints representing the standard order are stronger than the language-based constraints, and the understander goes with the stronger set of constraints.

In fact, it is difficult to really know how often task constraints play an important role in everyday comprehension. It seems reasonable to believe that they actually play a rather large role, and that tell-tale cases of conflict between task constraints and language constraints are just rare. This overriding of the language input, however, is far more frequent in the model than for the experimenter listening to the subjects. For this reason, it seems necessary to reduce the effect of the task constraints.

The question is what changes to the corpus are needed to switch the relative strength of the task and language constraints? One solution is suggested by the artificial corpus. Namely, reduce the strength of the task constraints by making the frequency of different commands similar. Reducing the regularities and increasing the combinatorics of the language in the corpus will force the model to learn strong language constraints. This solution strategy was pursued by St. John (1992) in a simulation study of text comprehension.

Another potential solution is to provide the model with a pre-training task in which there are no regularities aside from the language itself. The model would learn strong language constraints that it could then transfer to the puzzle task.

In general, a more combinatoric corpus with few semantic constraints produces essentially context free language constraints: each word means what it says and little more. Elman (1992) has suggested that the wide range of language contexts provided to children as they learn their native language serves to decorrelate the language from any specific context. This condition produces a virtually combinatoric corpus for children to learn, and underlies their ability to understand unexpected language input in the face of possible task constraints - just like the experimenter in the puzzle task. On the other hand, the relatively

weaker task constraints available in any given context can still facilitate comprehension when they cooperate with the language constraints.

Conclusions

The simulation of the natural corpus is preliminary and more work is required to make the model effective. In particular, some method must be found to change the relative strength of the language and task constraints. The strength of the task constraints does, however, demonstrate the powerful ability of the model to acquire and then use task constraints for comprehension.

More generally, constraint satisfaction provides a useful framework for learning and using constraints from different sources, whether lexical, syntactic, or task-oriented.

Constraint satisfaction is also a boon to processing informal language such as natural speech. The model does not attempt to match an input to a known grammatical form. Instead, all that is required of any input is that it contain sufficient constraints to compute the correct message, and those constraints can come either from the language or the task itself.

Using an easily represented task as the target of training provides other important advantages. First, it provides the technical advantage of relieving the experimenter of the burden of creating a linguistic theoretic representation of thematic case roles, propositions, or the like. The experimenter needs only to specify the task, and the model is required to learn whatever representation it needs to perform that task. This idea has the potential to significantly advance the science of PDP models of language comprehension.

A potentially limiting condition is the requirement of finding an adequate task for whatever language is desired to be learned. In this paper the language only pertained to referencing wooden blocks. However, with some creativity, the range of possible tasks and language may expand considerably.

One other advantage of using a task as the target of training is that the meaning of concepts and words do not have to be provided *a priori* in either the input or the target. The model acquires exactly those meanings necessary to perform the task. In this way the semantics of the language are grounded out in the task itself: language meaning as language use.

References

Allen, R. B., (1987). Several studies on natural language and back-propagation. *Proceedings of the International Conference on Neural Networks, Vol. 2*, p. 335-341, June 21-24, 1987, San Diego, CA.

Clark, H. H. (1985). Language use and language users. In G. Lindzey & E. Aronson (Eds.), *The handbook of social psychology, 2, (3rd ed.)*. New York: Harper & Row.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*, 179-212.

Elman, J. L. (1992). Personal communication.

Erickson, T. D. & Mattson, M. E., (1981). From words to meaning: A semantic illusion. *Journal of Verbal Learning and Verbal Behavior, 20*, 540-551.

Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic theory*. New York: Holt, Rinehart, & Winston.

Frederking, R. E. (1988). *Integrated natural language dialogue: A computational model.* Boston: Kluwer Academic Publishers.

Lehman, J. (1990). Adaptive parsing: Self-extending natural language interfaces. *Proceedings of the 12th Annual Meeting of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Miikkulainen, R. & Dyer, M. G. (1991). Natural language processing with modular PDP networks and a distributed lexicon. *Cognitive Science, 15*, 343-400.

Pollack, J. (1988). Recursive auto-associative memory: Devising compositional distributed representations. *Proceedings of the 10th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Reder, L. M., & Cleeremans, A. (1990). The role of partial matches in comprehension: The Moses illusion revisited. In A. C. Graesser and G. H. Bower (Eds.) *Inferences and text comprehension., The psychology of learning and motivation, vol. 25*. San Diego, CA: Academic Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1.* Cambridge, MA: MIT Press.

St. John, M. F. (1992). The story gestalt: A model of knowledge intensive processes in text comprehension. *Cognitive Science, 16*, 271-306.

St. John, M. F. & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence, 46*, 217-257.

Touretzky, D. & Hinton, G. E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of the 9th International Joint Conference on Artificial Intelligence.*