

Toward Connectionist Semantics

Garrison W. Cottrell¹
University of California at San Diego

Introduction

Much of the study of language has centered around the study of syntax, to the detriment of semantics and pragmatics. Part of the reason for this may be akin to the motivation of the besotted gentleman on his hands and knees beneath a streetlamp, who, when queried as to why he is looking on the sidewalk for the keys he lost in the alley, replies: "Because the light is better here!" I believe it is time to start mucking about in the alley; the keys are there. I also think we have a new flashlight: Parallel Distributed Processing². PDP mechanisms allow us to build machines whose fundamental operations include best fit search, constraint relaxation and automatic generalization. These are useful properties for processing language. I think the application of these models to NLP will change our view of what constitutes "semantics". I will argue that in order to deal with meaning seriously, we have to move beyond the folk-psychological level of symbols, and represent the microstructure of symbols. This is more than a granularity issue. It also has to do with the grounding of meaning in perception. It is on the level of microfeatures that I believe this grounding occurs, and PDP gives us a way to express this interface between language and perception.

My discussion of these issues will take the following course³. First I describe my previous work on word sense disambiguation in a PDP framework as a springboard for the rest of the discussion, and to give a sense of how lexical semantics might fit into an overall parsing model. Next I motivate a new model of word meanings through an example. I try to show that PDP has a natural way of expressing these meanings, and I give a sketch of how connectionist semantics could be learned. Finally, I briefly discuss metaphor.

Word sense disambiguation

One of the fundamental problems of natural language processing is word sense disambiguation. Determining the correct sense of a word for a particular use involves the interaction of many sources of knowledge: syntactic, semantic and pragmatic (i.e., "everything else"). In previous work (Cottrell, 1985) I have shown how word sense disambiguation can be modeled as a constraint relaxation process between competing hypotheses instantiated as nodes in a network representing linguistic knowledge. The representation is one that I have fancifully called *proclarative*: disambiguation happens as the result of activation spreading through a knowledge base where constraints between hypotheses are represented by positive and negative links between them. Figure 1 shows the basic structure of the model. The model operates as follows: First, words activate all of their lexical entries. These, in turn, activate syntactic and semantic (case) structures, which represent relations between word senses. It is feedback from these developing representations that provides support for the correct meanings and syntactic classes of the words. At the same time, bindings of constituents to roles in both syntax and semantics are mutually constraining one another to decide such things as prepositional phrase attachment. Thus parsing into a case structure is modeled as a three way constraint relaxation between the lexical entries of the words, the possible syntactic representations, and the possible semantic relations. Syntactic and semantic information are accessed in parallel, and operate simultaneously to determine the correct parse. This was

I would like to thank Mike Mozer, Harold Pashler, and Dave Rumelhart for helpful comments on this paper. Any haziness that remains is mine.

²I will assume familiarity with the connectionist, or PDP paradigm. The best introduction is Rumelhart and McClelland (1986).

³I will restrict myself here to lexical semantics. The generalization to logical form is left as an exercise for the reader.

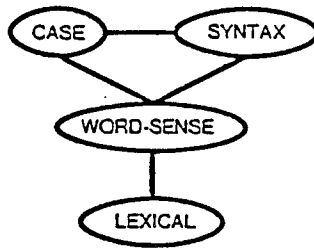


Figure 1. Sources of knowledge and constraint paths for disambiguation.

shown to be a useful model of the human disambiguation process, as evidenced by explanations of various psycholinguistic and neurolinguistic results.

One of the major weaknesses of that model was the representation of "meaning". Each meaning of a word is represented by a unit with an "awkward lexeme" (Wilks, 1976) as a label. Certainly, the label on a node is not important; it is the way the node connects up to other nodes that determine its relationship to other "meanings". But I think this is a general failing of almost all NLP programs currently in existence: the meaning of a word is best represented not as a symbol, but as an aggregate of connected microfeatures. I will next try to show why.

What is meaning? (A thought experiment)

It has been said that all words are polysemous to a degree. Let's take a fairly safe example: *truck*. This seems hardly polysemous, but it turns out we can bend the meaning, at least the image formed, in fairly continuous ways. Consider *Billy picked up the truck*. If you are like me, you get a picture of a small, probably plastic truck. In a symbolic system we might have a rule that if a usually large object is the object of a picking up action, then we should "toy-ify" it, either looking up the entry for "toy truck" or by applying a "toyification" transformation to the representation we had already retrieved: it weighs less, it is much smaller, it is composed of plastic. Of course, in *Superman picked up the truck*, we have an exception to the rule. And in *Bobby picked up the toy gun*, the application of the toy-ifying rule would need to be modified so that the size is not reduced. One can imagine that the list of rules and their application criteria might get a bit unwieldy.

One answer to this is, "Yes, the world is complicated." The problem is that this is not an isolated phenomenon. Rather, it pervades our conceptual landscape. The concepts that people use are not fixed entities, nor are they entities that vary discretely along a small number of dimensions. They *covary* in a continuous way. In *Tommy lugged the truck up the hill*, we imagine a heavier toy truck than the one Billy picked up, but a lighter one than Superman did. It might even be the same truck - *Billy picked up the truck and handed it to Tommy. Tommy lugged it up the hill.*" In this case it is *Tommy* that we imagine is smaller than *Billy*! Thus the interpretation we derive of the words in a sentence is the result of constraints between the meanings of the individual words, as well as the usual list: the structure of the sentence, the context in which it is spoken, the relationship between the speaker and the hearer, the shared knowledge, etc. People are very good at tasks like this that involve the application of multiple, simultaneous constraints. I claim that the "rules" that I attempted to describe above can emerge from the regularities of interaction among the internal structures of the concepts themselves, rather than an application of explicit rules to atomic concepts⁴. There is no reason that this could not be implemented in a "symbolic" system that has a constraint propagation mechanism, and continuous-valued levels of properties. The problem is that the modification would alter it so radically that we might as well have started with a connectionist model⁵.

⁴I am not claiming these are simply first order interactions; relations between feature clusters also need to be captured.

⁵Another reason for starting with a connectionist model is the existence of powerful learning algorithms that can derive constraints between features, as we will see below.

A modest proposal

In this section I will draw on previous work of others to lay out how a connectionist model can represent the kind of meanings that I think our experiment with truck point to. The basic idea is that meanings are connectionist schemata. These are assumed to be embedded in a system like the one I described above for word sense disambiguation - that is, they are getting input from other schemata concerned with syntax and larger semantic (case) structures.

Connectionist Schemata. Rumelhart et al. (1986) have demonstrated how a connectionist model of a schema can do something no implementation has done before: represent smoothly varying constraints between the slot fillers. The demonstration model represents the information we have about rooms. Each unit of the model represents one of forty possible descriptors and contents of a room: size, walls, ceiling, bathtub, stove, etc. The connection strengths between the units of the schema model were derived from people's reports of what they expected to find in each kind of room. (The weights were set according to the conditional probability that one item was reported given another item was reported.) Things that occurred together often were given a strong positive weight, things that never occurred together were given a negative weight. For example, every room has walls and a ceiling. These have a strong positive connection between them because they always co-occur. Probing the model consists of "clamping on" some units, which then activate positively connected units, and inhibit ones negatively associated with them. The office schema, for example, can be accessed by probing the model with "desk" (and "ceiling", to simulate the context is "room") (see Figure 2). The "prototype" rooms are shown to be peaks in a "goodness surface" in the space of unit activations that reflects the number of constraints satisfied between units of the model. The activation of the units travels up the goodness surface to the corner where the elements of the office schema become activated. This type of pattern completion is a typical way to access information in connectionist models.

An interesting variation on this is when two items are probed together that do not normally co-occur. For example, if the model is probed with "bed" and "sofa" what results is a large bedroom with a fireplace. The goodness space has been warped by these two inputs to form a new stable peak, where the filler of one of the slots, "size-of-room", has constrained what will be in the contents of the room in a way that is intuitively pleasing.

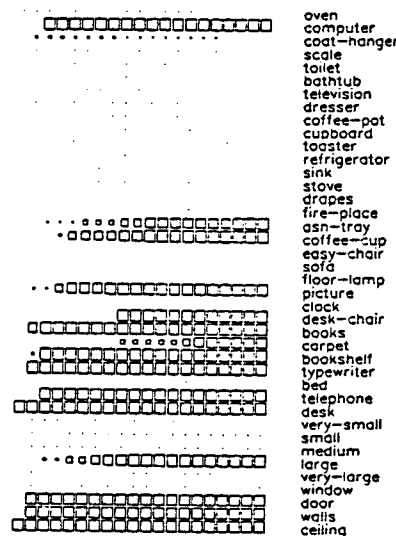


Figure 2. Probing the model with "desk" and "ceiling". The size of the square indicates activation value of the unit, and time moves from left to right. (From McClelland & Rumelhart, 1986, reprinted by permission).

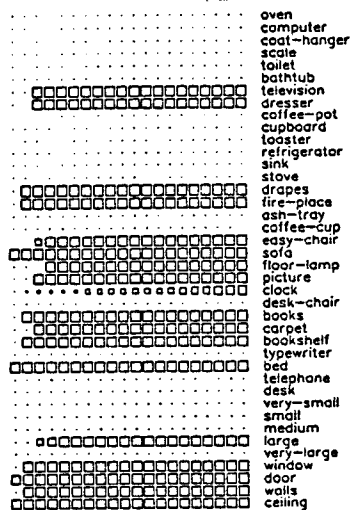


Figure 3. Probing the model with "bed" and "sofa". (From McClelland & Rumelhart, 1986, reprinted by permission).

It is possible to train a connectionist model to exhibit this blending of meanings, and to do so at the more micro-level I am advocating for word senses. McClelland & Kawamoto (1986) trained a network to assign case roles to nouns presented in a matrix as VERB-SUBJ-OBJ-MODIFIER. The representation of the input was a set of features for each syntactic slot that were linked to output feature schemata for each case role. The model was trained on a set of sentences in this format, and then tested on novel sentences. When given the novel sentence *the doll moved*, the model interpreted the doll as *animate*, because of the shared features between doll and humans and a tendency to assign animacy to agents. Thus, the model adjusted the meaning to fit the situation. The point is, distributed connectionist representations that represent symbols such as "doll" as a set of features and constraints can *relax* those constraints depending on external constraints - inputs from combinations of features in the schemata of the other words in the sentence.

These models assumed the elements of the schemata - the micro-features - were chosen by the modeler. The next section deals with how the features themselves might be learned, and how they might be grounded in perceptual processes.

Learning. A problem with any representation of meaning in terms of features is the infinite regression of features defined in terms of features. What is the basis clause of the inductive process of building a semantic representation? I believe that semantics must fundamentally be based in perception of and interaction with the environment. Powerful new algorithms have been discovered that allow connectionist networks to develop their own internal representations of their environment. Surprisingly, a rather useful network is one that does an identity mapping (Figure 4). The network has an input and output layer connected through a smaller layer of *hidden units*. By forcing the network to reproduce the input on the output through this narrow channel, it has to learn an efficient encoding of the input at the hidden unit layer. Such networks are self-organizing systems that learn to represent the important features of their environment.

These systems have been used to encode natural images and speech signals (Cottrell, Munro & Zipser, to appear; Elman & Zipser, 1986). The internal representations devised by these two systems (auditory and visual) can then be the "environment" to a third system which would take into account covariances between the two of them in a unified abstract encoding of

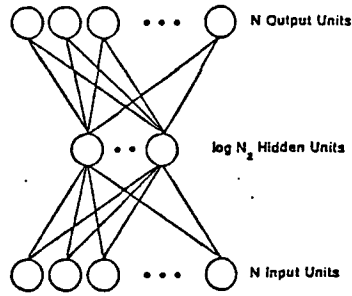


Figure 4. A network that develops an efficient encoding of its environment.

sound and light (see Figure 5)⁶. Now it will only take one of the input modalities to evoke the other. The input of an image would activate the image encoding, which in turn would partially activate the unified encoding of associated sounds and images. This can be filled out by pattern completion, enabling the unified encoding to feed back and activate the encoding of the word associated with the image. That is, an image will evoke a word and a word an image. While this is an oversimplified sketch, the important point is that connectionist systems use a uniform representation medium for both modalities, and thus afford the modeler an ease of communication between visual, proprioceptive and auditory inputs. Thus, this approach promises a computationally viable way to ground the infinite regress of meaning in associations between speech sounds with other perceptual representations generated from interactions with the environment. While this is just the base case of the induction, it has not been addressed by other approaches.

Metaphor. A second problem for a model of meaning is the question of metaphor. How could a connectionist system learn the metaphorical mappings that are such a big part of language? Connectionist schemata that have many stable states reflecting related meanings may account for much of what we call "metaphor". But how might new meanings be learned that are more radical transformations of old ones? For example, how might we learn that *I feel up today* means one's mood is elevated? Our identity mapping networks can be put to good use here in the following way⁷. Suppose we divide up the input pattern in Figure 4 into portions corresponding to a function, an input and an output. So the triple (F a b) represents $F(a) = b$, and given (F a b) the network produces (F a b). If we add a pattern completion network

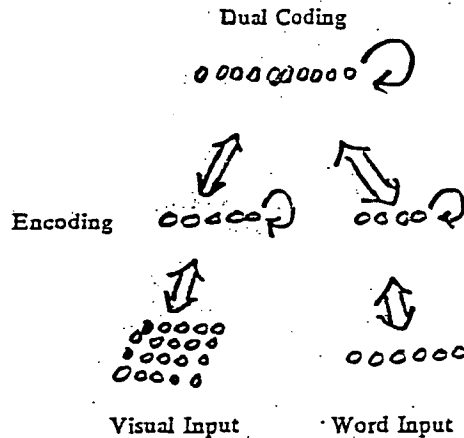


Figure 5. Automatically learned inter-modal encoding.

A similar idea has been independently proposed by Chauvin (1985).

⁷The following network is implemented, as McClelland would say, in "hopeware".

on the output layer, we can now give the network $(F a *)$ (where $*$ represents no input) and it will produce $F(a,b)$, computing that $F(a)$ equals b . In fact, within resource limitations, we can give it $F(*,b)$ or even $*(a,b)$ and have it invert the mapping or induce the relationship between the arguments. In ambiguous cases it will produce blends of the possible answers.

Now, assume that we have enough units in the argument positions that we can represent anything we want, and that we have trained it with functions and arguments from several disparate domains. Suppose we now give the network a function F with an argument c that is not in the domain of F . One characteristic of these networks is that they map similar inputs to similar outputs. The degree of overlap between the features of c and the features of elements of the domain of F will determine the coherency of the mapping. If c is sufficiently similar to a previously learned input, it will map c to an output similar to the previous one. It is able to do this because the mapping reflects constraints it has learned between the features of the inputs and outputs of F . If c is sufficiently different from other inputs it has learned in the domain of F , the result will be uninterpretable. Somewhere between these two is metaphor⁸.

Conclusion

I have attempted to show in this paper that word meanings are more of a moving target than we would like to think, and that they *covary* depending on constraints between them. The connectionist approach to semantics has a natural way to capture these smoothly varying constraints and meanings. I also have sketched how these meanings can be grounded in perceptual encodings and how some aspects of metaphor might be captured in this framework.

References

- Chauvin, Yves. (1986) Hypermnnesia, back-propagation, categorization, and semantics. Preprints of the Connectionist Models Summer School, Carnegie-Mellon University, Pittsburgh, Pa., June 21-29, 1986.
- Cottrell, G.W. (1985) A connectionist approach to word sense disambiguation. (PhD thesis) Available as TR 154, University of Rochester Computer Science Department.
- Cottrell, G.W., Munro, P. & Zipser, D. (to appear) Image compression by back-propagation. Technical Report, Institute for Cognitive Science, UCSD.
- Elman, J.L. & Zipser, D. (1986) Discovering the structure of speech. Paper presented at the 112th meeting of the Acoustical Society of America December 1986, Anaheim, Ca.
- McClelland, J.L. & Kawamoto, A.H. (1986) Mechanisms of sentence processing: Assigning roles to constituents. In J.L. McClelland and D.E. Rumelhart (Eds.) *Parallel Distributed Processing: Explorations in the microstructure of cognition*. Cambridge, MA:Bradford.
- Rumelhart, D.E. and McClelland, J.L. (1986) *Parallel Distributed Processing: Explorations in the microstructure of cognition*. Cambridge, MA:Bradford.
- Rumelhart, D.E., Smolensky, P.E., McClelland, J.L. & Hinton, G.E. (1986) Schemata and sequential thought processes in PDP models. In J.L. McClelland & D.E. Rumelhart (Eds.) *Parallel Distributed Processing: Explorations in the microstructure of cognition*, Vol. 2. Cambridge, MA:Bradford.
- Wilks, Y. (1976) Parsing English II. In Charniak and Wilks (Eds.), *Computational Semantics*. North-Holland, pp. 155-184.

⁸This represents a slight generalization of an idea of Dave Rumelhart's: his model did not include the function in the mapping.