A MICROFEATURE BASED SCHEME FOR MODELLING SEMANTICS

Lawrence A. Bookman
Computer Science Department
Brandeis University
Waltham, MA 02254
US.A.

ABSTRACT

One fundamental problem of natural language processing is word sense disambiguation. Solving this problem involves the integration of multiple knowledge sources: syntactic, semantic, and pragmatic. Recent work has shown how this problem can be modelled as a constraint satisfaction process between competing syntactic and semantic structures. We have defined and implemented a "locally-distributed" microfeature based model called MIBS, that uses a distributed short-term memory (STM) composed of microfeatures to represent the underlying sentence semantics. This work represents an improvement over previous work, as it provides a natural language understanding system a means to *dynamically* determine the current context and adjust its relationship with the sentences that follow. Here, the meaning of a word is represented not as a symbol in some semantic net, but as a collection of smaller features. The values of the microfeatures in STM vary *dynamically* as the sentence is processed, reflecting the system's "settling" in on the sentence's meaning. In addition they represent an *automatic* context mechanism that helps the system to disambiguate the sentences that follow.

I. INTRODUCTION

One fundamental problem of natural language processing is word sense disambiguation. Solving this problem involves the integration of multiple knowledge sources: syntactic, semantic, and pragmatic. Recent work (Cottrell & Small, 1983; Cottrell, 1985; Waltz & Pollack, 1985) has shown how this problem can be modelled as a constraint satisfaction process between competing syntactic and semantic structures. The above models rely primarily on local representations (one concept per node), as opposed to distributed representations, although Waltz and Pollack (1985) suggest ways that a static microfeature-based representation could be used to represent global contextual influences for competing word senses. Cottrell (1987) states that one of the major weaknesses of most NLP programs is their representation of meaning. Each meaning of a word is usually represented by a node with an "awkward lexeme" as a label, whereas its meaning is best represented not as a symbol, but as collection of variable valued microfeatures.

For a machine to "understand" language, it must have a means of "knowing" the context of a sentence and its relationship with the sentences that follow. Building upon the work of Waltz & Pollack (1985), we have defined and implemented a "locally-distributed" microfeature based model called MIBS (Microfeature Based Semantics), that uses a distributed short-term (STM) memory composed of microfeatures, to represent with more gradations of meaning, a sentence's underlying semantics. Our system provides a means to *dynamically* determine the current context and adjust its relationship with the sentences that follow. The model developed here, provides a basis for understanding this dynamic relationship and thus, this work presents a starting point from which one can build a more dynamic natural language understanding system. Note, Waltz & Pollack (1985) used a *static* microfeature-based representation, that had to be initially primed by a user. In addition, their system could not process multiple sentences in a continuous fashion, as their microfeatures were not dynamically updated.

Our microfeatures act as intermediate-level units (Minsky 1986) that are intended to capture underlying structural fragments, that is, pieces of semantic structure. These intermediate-level units represent intermediate concepts which may be indispensable for understanding language, because the comprehension of a complex sentence often hinges on composing a meaningful variation on a familiar theme. Although the microfeatures do not represent the relationships between the pieces of the semantic structure, they might provide a means for representing the underlying themes of the sentence and maybe an enriched notion of what the sentence "means". Thus, the microfeatures may be useful pieces for a cognitive model of language understanding. (See Bookman 1987 for details).

Our use of microfeatures is similar to Wilks' (1975) use of semantic primitives, in the sense that they are only partially definitional. In addition, we are using the microfeatures as a short-term memory, where currently active microfeatures influence the sentences to follow. This is similar to HInton's (1984) notion where each active unit in the network represents a "microfeature".

II. DESCRIPTION OF THE MODEL

The system is composed of two layers. The top layer is a "local" connectionist model, the bottom layer, a distributed layer of "microfeatures" (fig. 1). The "local" connectionist layer encodes the syntactic and lexical structure of the sentence (see section 4). While the microfeatures are used as a basis for defining the nodes (i.e. concepts, hypotheses) in the top layer, at least partially, and for associating each node with others that share its microfeatures. In addition they form the basis for our memory system. Each microfeature is potentially connected to every node in the top layer. Each node in the top layer is connected via bi-directional links to only those microfeatures that describe it. Collections of closely related nodes in the top layer will have many common microfeatures..........................

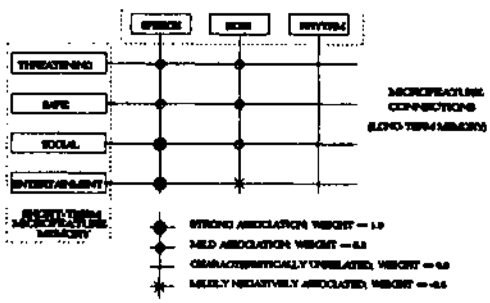

Figure 1. System Architecture

Given some initial setting it is the memory system (i.e., the distributed layer of microfeatures) that drives the system to eventually settle on the intended interpretation of the given sentence. Here memory is used in following limited sense: a short-term microfeature memory that stores the currently active and inactive microfeatures; and a long-term semantic memory that stores "semantic" knowledge in the form of connections between the microfeatures and concepts. This "semantic" knowledge is pre-wired into the network, and in the current implementation does not change. (Note, there is no "episodic" memory, as the model currently has no way of storing events). For example, in figure 1 the local node *speech* has a mild association with the microfeatures *threatening* and *safe,* and a strong association with *social* and *entertainment*

Here STM is represented by a microfeature vector, where each position of the vector corresponds to an independent microfeature, and the numerical value at that position corresponds to the level of activation of that feature. The initial activation of the microfeature vector primes local node concepts, and then the primed concepts change the activation of the microfeature values in STM, in turn activating new concepts (in this case the different word senses) in the top layer of the network. This is similar to Quillian (1968), except here successively activated (i.e. related) "concepts" are joined by (sub)bundles of microfeatures, rather than by single marker-passing links. The initial priming of memory can be set up by the experimenter, or can be automatically setup by the processing of a previous sentence. A major difference between this system and that of Waltz & Pollack (1985) is that this system performs relaxation on the microfeature set. As a by-product we get constellations of microfeatures that persist over time, that can be used to dynamically constrain the processing of the sentences that follow.

The basic computational units at each node are p-units with decay and p-units with conjunctive connections (for encoding semantic constraints). These units compute the potential (activation) of a node and are similar to the ones described in Feldman & Ballard (1982). In addition we are using meta-network structures called *network regions* for representing groups of competing word senses (Chun, Bookman & Afshartous 1987). One such example is the CN-REGION in figure 2. These structures provide more stable competition, are more tolerant to initial noise, and eliminate the premature "lock-in" effect of traditional WTA (winner-take-all) structures (Feldman & Ballard 1982). It is important to note that most of the computation here is local to each node and thus the relaxation algorithm can be performed by massively parallel hardware. Currently, the computation is performed on a Symbolics 3670, using a general purpose massively parallel simulator called AINET-2 (Chun 1986).

The heart of the algorithm consists of two steps: *node relaxation* followed by *microfeature relaxation. Node relaxation* computes the amount of activation a node is to receive from all its connecting neighbors, plus any activity from the currently active nodes of STM memory. The activity contributed by STM is computed by taking the product of a node's microfeature set (its "long-term semantic" knowledge) with the node's activation level, and, then taking the dot product of this result with the microfeature vector in STM memory. This result is then normalized, so that the relative contribution of the microfeature set is appropriately scaled. This computation allows the reactivation of previously active sets of nodes, and is similar in this regard to Minsky's "K-lines" (1980), where some agent (in our case the node) is associated with specific activated nodes (in our case the microfeatures), and this agent is used to recover the whole from any sufficiently large part.

The *microfeature relaxation* cycle computes the amount of activation each node contributes to the microfeatures currently active in STM memory. This is accomplished by taking the product of a node's microfeature set times the node's activation level (computed by the node relaxation computation above) and updating memory by adding the sum of all such computations to the old values in memory. The net effect is that evidence is accumulated, with memory being modified by recent experience.

By dividing the sum of the reiult by the turn of the absolute values of the microfeature values in STM.

Both of the above computations are performed at each simulation step for all nodes in the network.

Interestingly, the model proposed seems to correlate with some psychological processes, as evidenced by psycholinguistic data where subjects maintain the initial activation of all word senses and where a "post-access" decision takes place, following lexical access, that utilizes context to determine the appropriate meaning of the word (Swinney 1982). In summary, the computation models a constraint satisfaction process that uses the semantics of microfeatures to set context. The microfeature relaxation computation can be viewed as the post-access decision process that performs word sense disambiguation.

HI. THE MICROFEATURE SET

Below is a description of the microfeatures used in this research. We have tried to be somewhat systematic and fair in choosing them. We make no claims, however, that this is the only such set. Our purpose is only to demonstrate that microfeatures are useful. The features we chose can be broken down into the following categories:

- Lengths of events: second, minute, hour, day, week, month, year, decade.
- Temporal relationships between events: before, after, current.
- Locations: house, store, office, school, factory, casino, bar, restaurant, theatre, racetrack, city street, city park, city, rural, forest, lake, desert, mountain, seashore, canyon.
- Events: competition, social, business, entertainment.
- Distinctions needed to survive: threatening/safe, animate/inanimate, edible/inedible, good outcome/neutral outcome/bad outcome, moving/still, intentional/unintentional, inside/outside, temporary/permanent.
- Life themes: sleep, hunger, thirst, sex, sickness, health, death, crime, subsistence, marriage, education, learning, profession, work, hobby.
- Methods of communication: speech, written, machine, telephone, satellite.
- Means of transportation: walk, bus, car, train, airplane.
- Object site: very large, large, medium, small, very small.
- State change
- Goals: enjoyment goals, achievement goals, preservation goals, satisfaction goals. (See Schank & Riesbeck, 1981).
- Some primitives to encode event-types: atrans, mtrans, ptrans, propel. (See Schank & Riesbeck, 1981).

The categories: locations, events, methods of communication, means of transportation are important to a culture and were chosen on this basis. The categories: life themes, distinctions needed to survive, state change, length of events, and temporal relationships are common across cultures and were chosen for this reason. The size category te context dependent and provides a means for relational comparison. Goals and the Schank-inspired primitives were chosen as a basis for encoding "script-like" knowledge. There are roughly 90 microfeatures; however, probably several thousand are needed for a sufficiently rich semantics.

IV. SOME EXPERIMENTAL RESULTS

In the following examples we will illustrate the utility of the approach in determining the ''meanings" of the sentences:

(51) John went to Mary's party. He had a good time.

(52) John ran the 500 meters yesterday. He had a good time.

(53) John was talking to his boss. The language he used was inappropriate.

(54) John was programming at his computer. The language he used was inappropriate.

The networks that model these sentences are shown in figures 2 and 3 (actually only the local connectionist layer is shown). The networks for sentences S2 and S4 are not shown as they are similar to the ones for SI and S3 respectively. The rectangular nodes in the figure represent the input words. The elliptical nodes below them represent the competing word senses, and the structure above them the syntactic parse tree for the sentence.

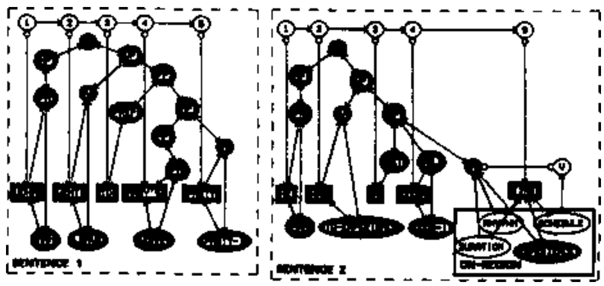

Figure 2. Network: "John went to Mary's party. He had a good time", after 30 cycles of relaxation.

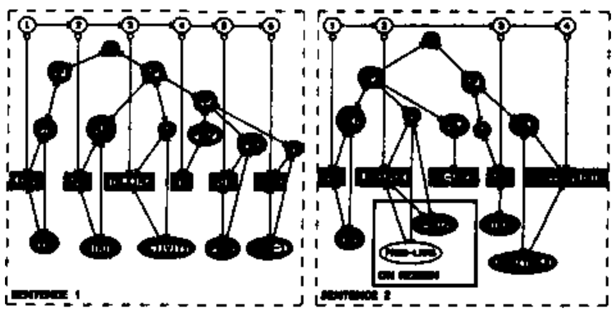

Figure 3. Network: "John was talking to his boss. The language he-used* was inappropriate", after 26 cycles of relaxation.

AUTOMATIC CONTEXT SETTING

In the first simulation, the two sentences (Si): "John went to Mary's party. He had a good time." were processed. Note, the meaning of the second sentence is ambiguous. In the context of *party* it means he had a good experience, but in the context of *track meet,* it

Note: we are using *hE USED* as as adjective to simplify tht parse tree. The correct parse could be represented as follows: np (Det (the in N| (g ^ Comp(|g he used Is NN means the measured time for the race he has just run was quite good. In this example we have used four senses of the input word "time":

RHYTHM - a noun, meaning the grouping of the beats of music.

DURATION - a noun, meaning the measured time for an event.

EXPERIENCE - a noun, meaning a person's experience during a specified period.

SCHEDULE - a verb, meaning to arrange or set the time of.

The purpose of this experiment was to demonstrate the resolution of the ambiguity of the second sentence, through the normal processing of the first sentence, but without any context being initially primed by an experimenter. Instead, this was to occur *automatically,* and *dynamically* through the constellations of microfeatures. After 30 cycles of processing, the system settled into the correct stable state, with the *experience* sense of *time* winning out (see fig. 2). Note we have sequenced the processing so that we first process the first sentence and then we process the second. If we process only the second sentence then the system will settle into an ambiguous state in which the two senses of *time: duration* and *experience* are both equally active. This demonstrates that the second sentence is really ambiguous to the system.

It is interesting to note that if the sentences were input in the reverse order (i.e. He had a good time. John went to Mary's party), the system still resolves the ambiguity. However, in this case it took the system 40 cycles of processing before it settled into the correct stable state. The increase in time correlates .with our own difficulty in trying to understand an ambiguous sentence that is out of context - it takes us a little longer also.

In the second simulation the two sentences (S2): "John ran the 500 meters yesterday. He had a good time." were processed. In this experiment we wanted to see if the system could correctly disambiguate the word *time* given the context of a different first sentence. After 26 cycles of relaxation, the system settled into the correct state, with the *duration* sense of *time* winning out. A plot (not shown) of the activation of the different word senses of *time* shows that the incorrect senses: *rhythm, experience,* and *schedule* quickly die out. As before, we have sequenced the processing so that we first process the first sentence and then we process the second sentence.

The sentences (S3) and (S4) were also tried and similar results were obtained. In either case the system was able to resolve the ambiguity of the second sentence, namely, John's manner of speech was inappropriate or the programming language he used was inappropriate. Figure 3 depicts the final state after having processed sentence S3. The processing of sentence S4 is similar, except here the *prog-lang* sense of *language* wins out.

V. SUMMARY

This paper has presented some preliminary results toward understanding the "meaning" of sequences of sentences. Its basic conclusions are that the semantics of microfeatures can be used to automatically and dynamically perform context setting as a sentence is being processed, helping to disambiguate the different word senses in the sentences that follow; and that the underlying sense of a sentence can be realized through the use of a distributed shared set of microfeatures. For a machine to "understand" language, it must have a means of "knowing" what the context is and what is its relationship with the sentences that follow. The model developed here provides a basis for understanding this dynamic relationship and thus, this work presents a starting point from which one can build a dynamic natural language understanding system. In addition, the approach presented allows for massively parallel networks to be used as the basis for implementing this mechanism.

Acknowledgements. Special thanks to Dave Waltz for suggesting many improvements to earlier drafts. Also special thanks to Hon Wai Chun for many discussions and comments on this paper, and James Pustejovsky for providing useful comments.

REFERENCES

1 Bookman, LA., "A Microfeature Based Scheme for Modelling Semantics," CS-1987-128, Computer Science Department, Branded University, Waltham MA 02254, 1087.

2 Chun, H.W., LA. Bookman, *k* N. Afshartous, "NETWORK REGIONS: Alternatives to the Winner-Take-All Structure,'* *IJCAI-87,* 1987.

3 Chun, H.W., "AINET-2 User's Manual," Computer Science Department, Brandeis University, CS-86-126, Waltham MA, 1086.

4 Cottrell, G.W. & S.L. Small, "A connectionist scheme for modelling word sense disambiguation," *Cognition and Brain Theory,* 6(1), (1083) 80-120

5 Cottrell, G.W. "A connectionist approach to word sense disambiguation," (PhD thesis) Available as TR 154, University of Rochester Computer Science Department, 1085.

6 Cottrell, G.W., "Toward connectionist semantics," *Proceedings of TINLAP-S,* 1087.

7 Feldman, J.A. and D.H. Ballard, "Connectionist Models and Their Properties," *Cognitive Science,* 6, (1082) 205-254.

8 Hinton, G.E., "Distributed Representations," CMU-CS-84-157. Computer Science Department, Carnegie-Mellon University, Pittsburgh PA 15213, 1084.

0 Minsky, M.L. "K-lines: A theory of memory," *Cognitive Science,* 4, (1080) 117-130.

10 Minsky, M.L. (draft) "Connection Machines, Parallel Learning, and Societies of Mind," 1086.

11 Quillian, M.R. "Semantic Memory," In M. Minsky (Ed.) *Semantic Information Processing,* Cambridge: MIT Press, 1068.

12 Schank, R.C. and C.R. Riesbeck, *Inside Computer Understanding,* Hillsdale, New Jersey: LEA, 1081.

13 Swinney, DA., "The structure and time-course of information interaction during speech comprehension: Lexical segmentation, access, and interpretation. In J. Mehler, E.C.T. Walker, and M. Garrett (Eds.), *Perspectives on mental representation.* Hillsdale, NJ: Erlbaum, 1082.

14 Walts, D.L. and J.B. Pollack, "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation/' *Cognitive Science,* 0, (1085) 51-74.

15 Wilks, Y, "Primitives and words," *Proceedings of TINLAP-1,* Cambridge, Ma. June 1075, pp. 42-45.