TECHNICAL REPORT NO. 275

Robust Lexical Selection in Parsing and Generation

by

Michael Gasser

April, 1989

COMPUTER SCIENCE DEPARTMENT

INDIANA UNIVERSITY

Bloomington, Indiana 47405–4101

Robust Lexical Selection in Parsing and Generation[1]

Michael Gasser

Abstract

A well-known difference between human language understanding and typical computational theories of language understanding is in the degree to which they handle partial or errorful input. While computational models tend to be brittle, people are remarkably good at interpreting linguistic input which deviates from the norm. In language generation there is an analogous problem, that of selecting an appropriate lexical entry when there is none in memory which matches the pragmatic/semantic input to generation. This paper presents a localized connectionist model of robust lexical selection for both language understanding and generation. Processing takes the form of pattern completion, where patterns consist of complexes of semantic, morphosyntactic, and pragmatic features. The system is presented with portions of such patterns and retrieves others. In generation what is given is pragmatic/semantic information and in understanding it is mainly morphosyntactic information. This approach is not only a natural way of accommodating both understanding and generation in the same system but it also permits the sort of robustness which is exhibited by human language processors.

1 Robustness in Parsing and Generation

A well-known difference between human language understanding and typical AI language understanding programs is in the degree to which the two systems handle partial or errorful input. AI systems tend to be brittle; when the input to a parser does not conform to any of the patterns stored in memory, the system breaks down. People, on the other hand, are remarkably good at interpreting linguistic input which deviates from the norm. The best-known examples of the robustness of the human parsing mechanism are utterances which are devoid of any syntax at all, yet which remain interpretable on purely semantic grounds:

`(1) mary paycheck receive go bank deposit`

Yet the phenomenon is more general that this. It includes on the one hand the handling of phonologically degraded input and on the other the identification and interpretation of lexical items used in non-standard ways. This paper is concerned primarily with the latter type of robustness. Consider the following sentence, a variant of one actually produced by a non-native speaker of English.

`(2) He deposited his property to his friend.`

The intention of the speaker was to describe a situation in which the actor leaves his valuables with a friend for safekeeping while he is away on business. The use of *deposit* in this sentence is decidedly odd, yet a native listener of English would have no trouble understanding it in context.

There are good reasons why people are robust understanders. The utterances which one encounters, in particular in spoken language, are often deficient in one way or another, and even when they are well-formed, the human perceptual apparatus is simply unable to pick up all of the relevant features.

Less attention has been paid to the corresponding ability in language generation. Consider the process of lexical selection, the central component in generation, though one that has not been the subject of much research (Levelt & Schriefers, 1987). Given a set of semantic/pragmatic features associated with a particular lexical item, it will in general not be the case that all of these features will be present in the input to the lexical selection process in generation. Rather <u>enough</u> of the features need to be present for the appropriate item to be

---

[1]The research reported on here was conducted in part while the author was at the AI Laboratory, UCLA. It was supported there by grants from the ITA Foundation and the JTF program of the U.S. Department of Defense.

selected. Thus in this sense the input to generation is normally incomplete, requiring robustness in the lexical selection process.

Lexical selection is made more difficult when input features conflict with features of lexical entries, that is, when there is no entry in memory which matches all of the features to be conveyed. This would correspond to the problem of having something to say but no completely appropriate way of saying it. Second language speakers often face such problems. In general, speakers seem again to be able to cope; they identify the lexical item which does the job better than any other. Sentence (2) above is a good example. The speaker may know *deposit* only in its sense of transferring money to a financial institution, but it still seems the best available word.

As is true for understanding, it seems the human language processor could not be otherwise. We are forced to map an infinite variety of potential topics of discourse onto a finite set of lexical items and structures. Of necessity the set of semantic/pragmatic features we choose to convey at any given time will rarely match those associated with particular lexical items. Again robustness is called for.

2 Parsing and Generation as Pattern Completion

There is general agreement on the desirability of accommodating both language generation and comprehension within the same system, though there is disagreement on the degree to which knowledge can be shared by the two processes. At the very least, people certainly learn most of what they know about generating language through the process of parsing language, and this fact implies a significant amount of sharing. The processes themselves, while largely the reverse of each other in terms of their inputs and outputs, can both be viewed as a form of **pattern completion**. That is, if linguistic patterns are seen as complexes of semantic, morphosyntactic, and sometimes pragmatic features, the processing system is given some of these and must retrieve others. In parsing what is given is mainly morphosyntactic, and what is retrieved is semantic and pragmatic, though available semantic and/or pragmatic features normally guide the process as well. In generation, the given features are mainly semantic and pragmatic and the retrieved features morphosyntactic. On this view, not only do both parsing and generation consist of the completion of incomplete patterns; both processes make use of the same patterns.

In certain connectionist models (Feldman & Ballard, 1982; Rumelhart, Hinton, & McClelland, 1986) the dominant mode of processing is pattern completion. In these models activation on a set of units representing a portion of a learned pattern tends to lead to the activation of the other units that take part in the pattern. Thus this approach is ideal for modeling parsing and generation within a single system. For parsing, processing is initiated through the activation of a set of units representing morphosyntactic and possibly also semantic or pragmatic features, and the output consists of the activation of units representing a complete set of semantic and/or pragmatic features. For generation processing is initiated through the activation of a set of units representing primarily pragmatic and semantic features, and the output consists of the activation (in the appropriate sequence) of units representing words. All associations within the network are bidirectional so that activation can spread in either direction.

Pattern completion in connectionist models is designed to find the best matching characterization in memory for a given set of input features. Thus it is admirably suited to the problem of coping with errors or missing information in the input to understanding and generation.

This paper discusses the robustness of lexical access in the **connectionist lexical memory** (CLM) model of sentence processing, a localized connectionist approach. The model is implemented in a program which generates and parses simple English sentences.

3 A Connectionist Framework for Parsing and Generation

The main features of the CLM model are the following:

1. Memory consists of a network of nodes joined by weighted connections. The system's knowledge is embodied entirely in these connections.

2. Concepts are represented as frames consisting of subnetworks of the memory.
3. The basic units of linguistic knowledge are subnetwork frames associating surface-level form directly with function. These form-function mappings comprise an inventory from which selections are made during generation and parsing.
4. Processing consists in the parallel spread of activation through the network starting with nodes representing inputs. The amount of activation spreading along a connection depends on the connection's weight and may be either positive (excitatory) or negative (inhibitory). Activation on nodes decays over time.
5. Decision making takes the form of competition among sets of mutually inhibiting nodes and the eventual dominance of one over the others.
6. Processing is more interactional than modular. Pragmatic, semantic, and syntactic information may be involved simultaneously in the selection of units of linguistic knowledge.

The system exhibits robustness in that it can find patterns to match input even when there are no perfect matches. Other aspects of human language processing which are modeled include 1) parallelism and competition, 2) priming effects, 3) a combination of top-down and bottom-up processing, 4) flexibility in generation, 5) speech errors involving substitution, and 6) knowledge in the form of tendencies with degrees of associated strength rather than strict rules or constraints. In addition, the model accommodates both generation and parsing. An earlier version of the CLM model is described in detail in Gasser (1988).

3.1 Linguistic Memory

Memory in the model is a localized connectionist implementation of a semantic network similar to Fahlman's NETL (1979). In NETL, roles (slots), such as ACTOR, COLOR, and SUBJECT, take the form of nodes rather than links, and links are confined to a small primitive set representing in particular the IS-A, HAS-A, and DISTINCTNESS relations. In the present model, these links are replaced by pairs of weighted, directed connections of a single type, one connection for each direction.

Linguistic knowledge is integrated into the rest of memory. The basic units of linguistic knowledge are generalizations of two types of acts: **illocutions** and **utterances**. In this paper I will be mainly concerned with the latter. A generalized utterance, or **entry**, is a frame (implemented as a network fragment) associating a morphosyntactic pattern with a set of semantic and possibly also contextual features. Entries include frames for clauses, noun phrases, adjective phrases, and prepositional phrases. They are arranged in a generalization hierarchy with syntactic structures at its more general end and phrasal lexical entries at its more specific end. Thus lexical entries in the model are just a relatively specific type of entry. An entry normally has a node representing the whole phrase, one or more nodes representing constituents of the phrase, and one or more nodes representing semantic or pragmatic aspects of the phrase.

Figure 1 shows a portion of the lexical entry for the verb *deposit* in the sense of 'leaving money in a bank'. Nodes are denoted by rectangles and pairs of connections by lines. For convenience frame boundaries are indicated by fuzzy rectangles with rounded corners, but these boundaries have no significance in processing. Node names likewise are shown for convenience only; they are not accessible to the basic procedures. Names of lexical entries begin with an asterisk, and lower-case names indicate roles. The lexical entry shown in the figure, *DEPOSIT-IN-BANK, associates clauses having a form of the word *deposit* as their main verb with instances of the concept of ABSTRACT-TRANSFER having MONEY as their OBJECT and BANK as their DESTINATION. The frame is represented as a subtype of *ABSTRACT-TRANSFER, the general frame for clauses referring to instances of ABSTRACT-TRANSFER. Other subtypes of this frame include entries such as *GIVE, *SEND, and *STEAL.

Note that the *DEPOSIT-IN-BANK entry includes the information needed to associate semantic and synactic roles. For example, there is a connection joining the SUBJECT[2] constituent with the ACTOR of the instance of ABSTRACT-TRANSFER that is being referred to. Likewise the DIRECT-OBJECT and IN-CONSTITUENT (the constituent with *in* as case marker) are associated with the appropriate semantic roles. In the *ABSTRACT-TRANSFER entry only one constituent is shown in the figure, the TO-CONSTITUENT[3] which refers to the RECIPIENT of the transfer. Note how this is linked to the corresponding roles in the *DEPOSIT-IN-BANK entry.

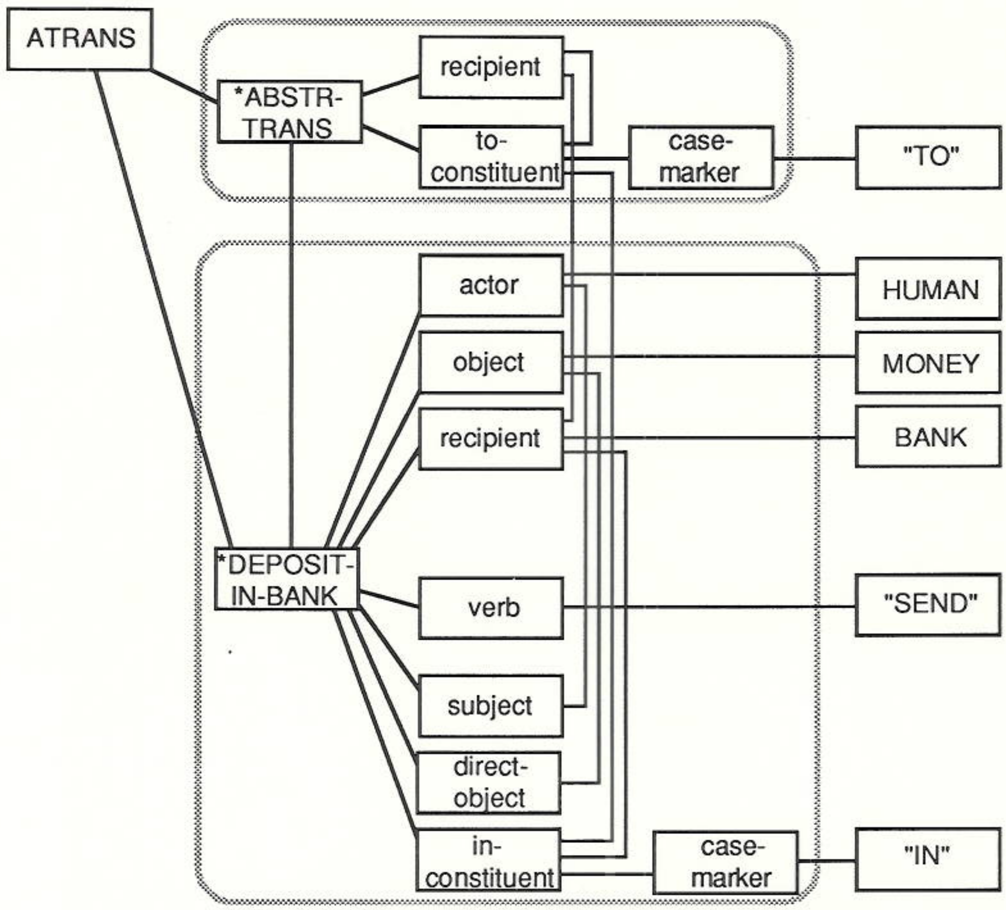

Figure 1: Portion of the entry for *depositi*ng money in a bank

3.2 Processing in General

Each node in the network has at any given time an activation level. When the activation of a node reaches its threshold, the node fires and sends activation along all of its output connections. The firing of a node represents a decision made by the system. For example, the selection of an entry matching an input pattern is represented by the firing of the head node of the entry. Following firing, a node is inhibited for an interval during which its state is unaffected by inputs from other nodes. After this interval has passed, the node recovers with a small amount of positive activation and can be further activated from other nodes.

The amount of activation spreading from one node to another is proportional to the weight on the connection from the source to the destination node. The weight may be high

---

[2] In its present form the entry applies to active clauses only. For simplification I have ignored the possibility of passives.

[3] The system does not have knowledge of indirect objects, that is, constituents referring to the RECIPIENT (or BENEFACTOR) that take no case marker.

enough to cause the destination node to fire on the basis of that activation alone. For example, when activation spreads along a connection from an instance to a type node, say, from JOHN to HUMAN, we generally want the type node to fire immediately. In most cases, however, activation from more than one source is required for a node to fire. Connection weights may also be negative, in which case the relationship is an inhibitory one because the negative activation spread lessens the likelihood of the destination node's firing.

Sometimes we want only one node from a set to fire at a given time. For example, in the generation of a clause, the system should select only one of the set of verb lexical entries. In such cases the members of the set form a network of mutually inhibiting nodes called a **winner-take-all (WTA) network** (Feldman & Ballard, 1982). When two or more of these nodes receive activation, a process is initiated by which nodes with more activation in effect draw activation from those with less. This usually results eventually in the firing of one of the nodes, at which point the winner-take-all process terminates.

The model also has a decay mechanism reflecting the importance of recency in processing. The activation level of all nodes decreases at a fixed rate.

3.3 Language Processing

Language processing can be viewed as a series of selections, each made on the basis of a set of factors which make quantitative contributions to the decisions. During sentence generation the items selected include general morphosyntactic patterns for the sentence and its constituents (e.g., STATEMENT, COULD-YOU-QUESTION, COUNTABLE-NP, etc.) and a set of lexical items to fill the slots in these patterns. During sentence analysis the items selected include word senses, semantic roles to be assigned to referents, and intentions to be attributed to the speaker.

In the CLM model the selection process is implemented in terms of 1) the parallel convergence of activation on one or more candidate nodes and 2) the eventual dominance of one of these nodes over the others as a result of mutual inhibition through a WTA network. Consider the case of lexical selection in generation. Activation converges on a set of candidate lexical entries starting from nodes representing conceptual features of an input. Any number of entries may receive some activation for a given input, but because the entries inhibit each other through a WTA network, only one is selected.

Input to generation consists of a set of firing nodes representing a goal of the speaker. As activation spreads from the input nodes, it converges on nodes representing a general pattern appropriate for the goal type, for example, the STATEMENT pattern, and a set of patterns appropriate for the propositional content of the goal. These include lexical patterns such as *DEPOSIT-IN-BANK and *MONEY as well as grammatical patterns such as PAST-CLAUSE and INDEFINITE-NP.

The same basic mechanism works for parsing. Input consists of firing nodes representing words. These are given to the program at intervals of four time steps. Activation from the word nodes converges on entries for lexical and syntactic patterns. For ambiguous words there are two or more entries which inhibit one another through a WTA network. Lexical selection leads to the firing of conceptual nodes representing the interpretation of the input.

Alongside entry selection, the basic processing mechanism also implements the temporary **role binding** that is necessary for both generation and parsing and the appropriate output **sequencing** of constituents that is required for generation. These aspects of processing are not discussed further in this paper; for details, see Gasser (1988).

4 An Example

4.1 Parsing

Consider first the selection of the verb lexical entry that takes place during the parsing of sentence (2). A more complete version of the entry *DEPOSIT-IN-BANK is shown in Figure 2. Here an abbreviated notation is used. Connections between head and role nodes are represented by indentation, and connections among roles within the same frame are represented using a caret preceding the name of the role at the other end of a connection. Thus the SOURCE role in *DEPOSIT-IN-BANK is connected to the ACTOR role within the same frame. The dotted line separates semantic from morphosyntactic features within the entry.

The system also has two other entries for the verb *deposit*, one meaning 'the PHYSICAL-TRANSFER of MONEY into a VENDING-MACHINE', the other 'the PHYSICAL-TRANSFER of PARTICULATE-MATTER onto a HORIZONTAL-SURFACE through the action of some NATURAL-FORCE'.

```
(*DEPOSIT-IN-BANK *ABSTRACT-TRANSFER ABSTRACT-TRANSFER
     (actor HUMAN)
     (object MONEY)
     (source ^actor)
     (recipient BANK)
     (duration TEMPORARY)
     (purpose1 PREVENT                ;;keep money safe
         (object LOSE
              (object ^object)))
     (purpose2 INCREASE-VALUE         ;;earn interest
         (object ^object))))
-------------------------------------------------------
     (verb "DEPOSIT")
     (subject ^actor)
     (direct-object ^object)
     (in-constituent ^recipient
         (case-marker "IN")))
```

Figure 2: Entry for *deposit*ing money in a bank

During the parsing process, nodes representing both conceptual and formal features of the input are activated, and these in turn activate roles in various entries as paths of activation intersect there. For sentence (2), the parsing of the subject results in the firing of the ACTOR and SUBJECT roles in *DEPOSIT-IN-BANK as well as the corresponding roles in all entries for which there is a SUBJECT referring to a HUMAN ACTOR. The recognition of the verb causes the VERB role to fire in those three entries which have *deposit* in this slot. The appearance of the direct object, however, does not strongly activate the DIRECT-OBJECT and OBJECT roles in any of these entries because in sentence (2) the direct object refers not to money or particulate matter but to some unspecified property. The same is true for the IN-CONSTITUENT. In this case the input deviates in two ways from what appears in the *DEPOSIT-IN-BANK entry: the referent is not a BANK, and the CASE-MARKER in the constituent is not "IN".

At this point, the most strongly competing entries are *DEPOSIT-IN-BANK and *DEPOSIT-IN-VENDING-MACHINE. In both cases the entry head nodes receive activation from their SUBJECT/ACTOR roles (because they are linked to HUMAN) and from their VERB roles (because they are linked to "DEPOSIT"). Here we can imagine a variety of information that a human understander might use in selecting one of the two lexical entries over the other. This system is currently not sophisticated enough to make elaborate inferences, and it needs some help from the context. In this case, we assume that the context has led the system to expect that the actor will want to protect his property. This leads to the firing of the PURPOSE1 role in *DEPOSIT-IN-BANK, providing the extra activation that this entry needs to win out over *DEPOSIT-IN-VENDING-MACHINE. Figure 3 illustrates the competition between *DEPOSIT-IN-BANK and *DEPOSIT-IN-VENDING-MACHINE. The fuzzy line denotes an inhibitory connection, the thick black borders firing nodes, and the arrows the path of activation spread.

Once the head node for *DEPOSIT-IN-BANK has fired, it leads to the firing of more general entries associated with this one, in particular, *ABSTRACT-TRANSFER. *ABSTRACT-TRANSFER contains the information that the constituent referring to the RECIPIENT is normally marked with *to*. The firing of the node for this constituent provides the extra activation needed for the RECIPIENT role in *DEPOSIT-IN-BANK to fire, allowing the system to recognize that the friend referred to in the phrase following *to* is the intended recipient of the property. For details on how this temporary role binding process works, see Gasser (1988).

4.2 Generation

Consider now the generation of sentence (2). Assume the speaker has an entry like that shown in Figure 2 except that she does not know that the appropriate case marker for this use of *deposit* is *in*.

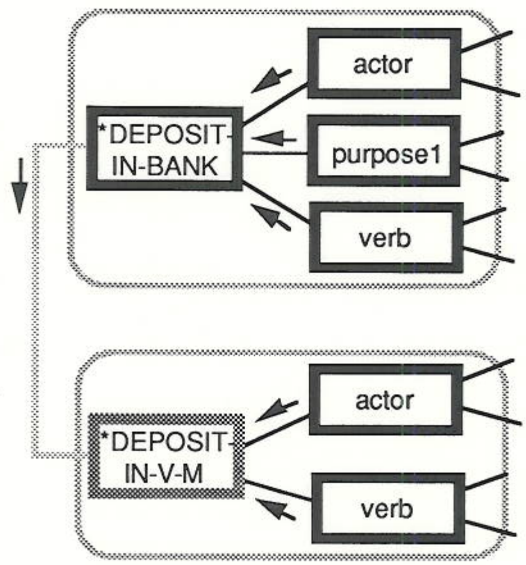

Figure 3: Competition between two entries during parsing

At the point in generation where the verb entry selection is to take place, a number of nodes representing the input concept will have fired. As activation spreads from these nodes, it will intersect on the roles of some entries. Within *DEPOSIT-IN-BANK, the ACTOR, SOURCE, RECIPIENT, DURATION, and PURPOSE1 (protecting the transferred property) roles will fire, while the OBJECT role, which is an instance of PROPERTY rather than MONEY, and the PURPOSE2 (earning interest) role, which is not applicable in this case, will not fire. The firing roles send activation to the head node of the entry, which also receives activation from ABSTRACT-TRANSFER. The head node competes via a WTA network with other verb lexical entries, including some which will also have significant activation, such as the entry for *give*. The WTA network sees to it that only one of the entry head nodes fires.

The *GIVE entry, like *DEPOSIT-IN-BANK, is associated with the general notion of ABSTRACT-TRANSFER. Like *DEPOSIT-IN-BANK, it includes the information that the ACTOR and SOURCE of the transfer are the same, but for *GIVE there is a tendency for the DURATION of the transfer to be PERMANENT. For the example, *GIVE receives input from ABSTRACT-TRANSFER and from the ACTOR and SOURCE roles. *DEPOSIT-IN-BANK, on the other hand, receives input from its RECIPIENT and DURATION roles (and others not shown in the figure) in addition to ABSTRACT-TRANSFER and the ACTOR and SOURCE roles. *DEPOSIT-IN-BANK wins out over *GIVE because of the activation received from a greater number of matching roles exactly as this happens among competing entries during parsing.

5 Related Work

Work on robust parsing within traditional symbolic frameworks has focused on problems of unfamiliar lexical items (e.g., Zernik, 1987) or ungrammaticality (e.g., Fain, Carbonell, Hayes, & Minton, 1985). Within the connectionist paradigm, McClelland & Kawamoto (1986) demonstrated that connectionist models are well suited to the problem of mapping syntactic to semantic case in the presence of novel verbs or missing arguments. In this paper, the focus has been on accessing entries for known lexical items, given input that deviates from that found in the entries. While the model requires refinement and further testing before it can be shown to handle all of the categories of input that these other approaches do, it is felt that these other types of processing difficulties will be accommodated within the single general framework proposed here.

Robustness in generation, on the other hand, has not come up in the literature. The generation work most closely related to the present approach is that of Kukich (1987) and Ward (1988). Kukich trained a connectionist network to associate semantic features with phrasal idioms. While she did not specifically address the issue of robustness, it is clear that her system would be able to handle noisy or incomplete input. However, the fact that there are no roles in her model makes the representation of constituency in concepts and patterns unwieldy if not impossible. Ward uses a spreading activation approach to model generation as a creative process. Thus, as in this paper, he is concerned with the fact that there are often no simple mappings between sets of input features and lexical entries. However, he does not make use of the competition among entries that seems to be required to deal with input features that do not match any entry in a straightforward way.

Unlike the current approach, none of the models discussed here handles lexical selection in both parsing and generation.

6 Conclusions and Future Work

In this paper I have characterized robustness as a general feature desirable in both language understanding and generation systems and have described a model which handles lexical selection in terms of the general mechanism of pattern completion. This mechanism not only provides a natural way of treating understanding and generation as similar sorts of processes operating on the same memory; it is ideally suited to coping with input that does not correspond precisely to the patterns stored in linguistic memory.

Current work is concerned with transforming the localized memory of the CLM model to a distributed memory. The advantages of distributed representations include a more efficient use of memory, tolerance to damage to memory, and more direct interaction among the relevant features, one that is not mediated by head nodes and by the levels in is-a hierarchies.

References

Fahlman, S. E. (1979). *NETL: A system for representing and using real-world knowledge.* Cambridge, MA: MIT Press.

Fain, J., Carbonell, J. G., Hayes, P. J., & Minton, S. N. (1985). MULTIPAR: A robust entity-oriented parser. *Seventh Annual Conference of the Cognitive Science Society*, 110-119

Feldman, J. A., & Ballard, D. H. (1982). Connectionist models and their properties. *Cognitive Science, 6,* 205-254.

Gasser, M. (1988). *A connectionist model of sentence generation in a first and second language.* (Technical Report UCLA-AI-88-13). Los Angeles: University of California, Los Angeles, Computer Science Department.

Kukich, K. (1987). Where do phrases come from: Some preliminary experiments in connectionist phrase generation. In G. Kempen (Ed.), *Natural language generation* (pp. 405-421). Dordrecht: Martinus Nijhoff.

Levelt, W. J. M., & Schriefers, H. (1987). Stages of lexical access. In Kempen (Ed.), *Natural language generation.* (pp. 395-404). Dordrecht: Martinus Nijhoff.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel Distributed Processing. Explorations in the microstructures of cognition: Vol. 2: Psychological and biological models* (pp. 272-325). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for Parallel Distributed Processing. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the microstructures of cognition: Vol.1: Foundations* (pp. 110-149). Cambridge, MA: MIT Press.

Ward, N. (1988). Issues in word choice. *Twelfth International Conference on Computational Linguistics*, 726-731.

Zernik, U. (1987). *Strategies in language acquisition: Learning phrases from examples in context* (Technical Report UCLA-AI-87-1). Los Angeles: University of California, Computer Science Department.