

# Sequencing in a Connectionist Model of Language Processing<sup>1</sup>

Michael GASSER<sup>2</sup>  
Michael G. DYER

AI Laboratory  
Computer Science Department  
University of California  
Los Angeles, California 90024, USA

## Abstract

Recent research suggests that human language processing can be profitably viewed in terms of the spread of activation through a network of simple processing units. Decision making in connectionist models such as these is distributed and consists in selections made from sets of mutually inhibiting candidate items which are activated on the basis of input features. In these models, however, there is the problem, especially for generation, of obtaining sequential behavior from an essentially parallel process. The thrust of this paper is that sequencing can also be modelled as a process of competition between candidates activated on the basis of input features. In the case of sequencing, the competition concerns which of a set of phrase constituents will appear in a particular output position. This account allows output ordering to arise out of the interaction of syntactic with semantic and pragmatic factors, as seems to be the case for human language generation. The paper describes a localized connectionist model of language generation, focusing on the representation and use of sequencing information. We also show how these same sequencing representations and mechanisms are usable in parsing as well.

## 1. The Problem of Sequencing in Generation

The order in which the constituents of an utterance appear depends on two kinds of factors: language-specific conventions and more or less universal tendencies. Examples of conventions are the placement of relative clauses after nouns in English and the reverse ordering in Japanese. Some of these conventions are absolute: relative clauses always follow nouns in English. Others are only tendencies and can be overridden. For example, in English direct objects usually follow verbs, but they may also come at the beginnings of clauses. Universal tendencies include in particular the appearance relatively early in a clause of material which is primed in some way (Bock 1982). Such psychological considerations may be the sole factor determining an item's position, as often happens in languages with relatively free word order such as Russian. But they also come into play when there is a linguistic sequencing convention which is a tendency rather than an absolute constraint.

Consider the case of the position of the arguments in ditransitive sentences in English. These sentences generally refer to an instance of some kind of transfer from one person to another. In such sentences the argument referring to the semantic OBJECT may precede or follow the argument referring to the RECIPIENT of the transfer. Other things being equal, if one of these arguments refers to something which has been mentioned recently, it will tend to come first. This tendency explains the strangeness of sentences (1b) and (2b).

(1a) *Instead of calling John, Mary sent him a letter.*

?(1b) *Instead of calling John, Mary sent a letter to him.*

(2a) *Instead of throwing away the letter, Mary sent it to John.*

?(2b) *Instead of throwing away the letter, Mary sent John it.*

One way to view this variation is in terms of competition between the two arguments to fill the position following the verb. One argument may have a head start if it has been primed in some way, in particular if its referent has just been mentioned. In the example sentences, the givenness of the referent results both in the priming that

leads that NP to come first and in the realization of the NP as a pronoun rather than a full noun phrase.

This explanation, in terms of competition for output positions, can account for other types of constituent order variation as well. An example is the alternative orders possible with transitive verb-plus-particle combinations in English: *take out the trash, take it out.*

Thus sequencing is a phenomenon involving competition and quantitative tendencies rather than absolute constraints. These properties make it reasonable to deal with sequencing within the framework of connectionist models, which we discuss in the next section.

## 2. Connectionism and Language Processing

In recent years there has been increasing interest in cognitive models built on networks of simple processing units which respond to the parallel spread of activation through the network (Feldman & Ballard 1982, McClelland, Rumelhart, & the PDP Research Group 1986). In the area of natural language processing, these models, generally referred to as **connectionist**, have been shown to exhibit interesting properties not shared by more conventional symbolic approaches. In particular, connectionist approaches to language analysis (e.g., Cottrell & Small 1983, McClelland & Kawamoto 1986, Waltz & Pollack 1985) are able to model priming effects and the interaction of different knowledge sources in lexical access. There have been only limited attempts to apply connectionist models to language generation (e.g., Dell 1986, Kukich 1986) but the potential there is also clear. While generation is usually conceived of as a top-down process involving sequential stages, it also involves bottom-up aspects, a good deal of parallelism, and "leaking" between the various stages, in addition to the priming effects which are handled well by spreading activation approaches.

Still, there are significant problems to be surmounted when treating language processing in a connectionist framework. An important one is the representation and utilization of information about the sequencing of constituents. While information about serial order is certainly a key element in parsing, it has been possible in existing connectionist parsing schemes to avoid dealing with this problem because of the limited sets of examples that are treated. Generation is another matter: no sentence can be generated without attention to the ordering of constituents. If connectionism is to succeed as an approach to human language processing, it must be possible to handle this sort of information within the confines imposed by the framework. This paper presents a localized connectionist model of language generation in which sequencing is dealt with in terms of basic features characteristic of these models: spreading activation, firing thresholds, and mutual inhibition. The same sequencing information is also usable during parsing. Most importantly, the approach offers a psychologically plausible account of sequencing in which syntactic and semantic factors interact to yield a particular ordering. The model is implemented in a program called CHIE which has been used to test the model's adequacy for a limited set of English and Japanese structures.

## 3. A Framework for Connectionist Language Processing

In this section we give an overview of knowledge representation and processing in the model. The main features of the 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 schemas consisting of subnetworks of the memory.
3. The basic units of linguistic knowledge are schematic subnetworks associating form directly with function. These form-function mappings comprise an inventory from which selections are made during generation and parsing.
4. Formally, the linguistic units are composed of surface-level patterns ranging from phrasal lexical patterns to purely syntactic patterns.
5. 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.
6. Decision making in the model takes the form of competition among sets of mutually inhibiting nodes and the eventual dominance of one over the others.
7. Processing is more interactional than modular. Pragmatic, semantic, and syntactic information may be involved simultaneously in the selection of units of linguistic knowledge.

The model provides a better account of human language generation than other computational models. In particular, it offers these advantages:

1. Parallelism and competition, which characterize human language generation, are basic features of the model.
2. Priming effects are naturally accommodated. Nodes are primed when there is activation remaining on them as a result of recent processing, and priming disappears as activation decays.
2. The system exhibits robustness in that it can find patterns to match conceptual input even when there are no perfect matches.
3. The approach allows for a combination of top-down (goal-driven) and bottom-up (context-driven) processing.
4. Generation in the model is flexible because spreading activation automatically finds alternate ways of conveying particular concepts.
5. Linguistic and non-linguistic knowledge take the form of tendencies with degrees of associated strength rather than strict rules or constraints.

The 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, semantic network 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 we will be mainly concerned with the latter. A generalized utterance (GU) is a schema (implemented as a network fragment) associating a morphosyntactic pattern with a semantic content and possibly contextual factors. GUs include schemas 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 GU. A GU 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 how a lexical entry would be represented in a simplified version of the system which does not incorporate information about sequencing. Nodes are denoted by rectangles and pairs of connections by lines. For convenience schema 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. Lower-case names indicate roles, and role names preceded by a colon are abbreviations of longer names. In the figure, for example, ":content" represents the CONTENT of \*SEND-MAIL. The lexical entry shown in the figure, \*SEND-MAIL, represents clauses with a form of the word *send* as their main verb, the concept of ABSTRACT-TRANSFER as their CONTENT, and MAIL as the MEANS of the transfer. The schema is represented as a subtype of the general schema for clauses, from which \*SEND-MAIL implicitly inherits other information (not shown in the figure).

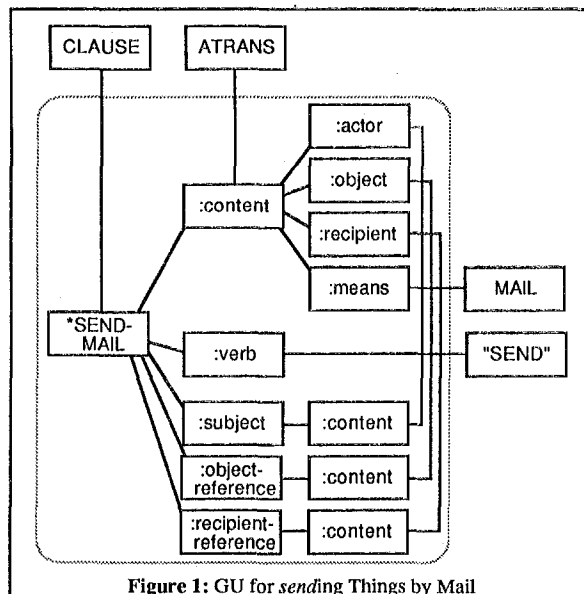


Figure 1: GU for sending Things by Mail

Note that the \*SEND-MAIL entry includes the information needed to associate semantic and syntactic roles. For example, there is a connection joining the CONTENT of the SUBJECT<sup>3</sup> constituent with the ACTOR of the CONTENT of the whole clause, that is, the person performing the instance of ABSTRACT-TRANSFER that is being referred to. The other two constituents shown represent the noun phrases referring to the semantic OBJECT and the RECIPIENT of the ABSTRACT-TRANSFER. The former could also be referred to as the "direct object" of the clause. The latter is realized either as an "indirect object", as in *Mary sent John the letter*, or a prepositional phrase with *to*, as in *Mary sent the letter to John*.

### 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 firing 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 a schema matching an input pattern is represented by the firing of the head node of the schema. 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 retains 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 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.

To simulate parallelism, the process is broken into time steps. During each time step, activation spreads from each firing node to the set of nodes directly connected to it. (In some cases activation may continue to spread beyond this point.)

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). The nodes are activated through the firing of a source node which is connected to all of the network members. At this time one of the network member nodes may already have enough activation to fire. If not, a specified interval is allowed to pass and if none of the members has yet fired, they receive additional activation, which is usually enough to cause one of them to fire. In any case, when one of the nodes fires, it immediately inhibits the others, effectively preventing them from firing for the time being.

### 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 present 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. All lexical entries, such as \*SEND-MAIL above, have a CONTENT role, and it is through this role that entries are selected during generation. Activation converges on the CONTENT role of a lexical entry starting from nodes representing conceptual features of an input. Any number of lexical entries may receive some activation for a given input, but because the CONTENT roles of 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 \*SEND-MAIL and \*LETTER as well as grammatical patterns such as PAST-CLAUSE and INDEFINITE-NP.

While some important aspects of parsing have not yet been implemented in CHIE, the basic mechanism works for parsing as well as for generation. 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 definite noun phrases, this leads to the firing of nodes representing referents. Verb entries specify the general proposition types and also provide for temporary "role binding". Role binding amounts to the firing in close proximity of a node or set of nodes representing a referent and a node representing its semantic role in the proposition. However, the program, like most other connectionist models, currently has no way of storing these role bindings in long-term memory.

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.

### 4. Sequencing

It is not a straightforward matter to implement sequential behavior within the confines of a system consisting of simple processing units that are activated in parallel. Alongside the basic problem of creating emergent sequential behavior from a parallel process, there is the need for sequencing information of two types to be

transmitted. When it is time for a constituent to be produced, it needs to signal its own daughter constituents to be produced in the appropriate sequence and, when these are completed, to signal sister constituents which follow it to be produced.

The thrust of this paper is that sequencing can be modelled like the rest of language processing, that is, as a series of selections made on the basis of interacting quantitative factors. Consider first how the parallel activation spread is turned into a sequential process during generation. Activation spreads initially from nodes representing the semantics and pragmatics of the utterance to nodes representing the lexical and grammatical patterns to be used, but the thresholds of the constituent nodes of these patterns are such that the nodes cannot yet fire. They fire only when they have received additional activation along connections specifying sequencing relations between constituents. When more than one constituent may follow a given constituent, there are connections to all of the alternatives. The weights on these connections represent degrees of syntactic expectation regarding which constituent will follow, and the constituent nodes inhibit each other through a WTA network which permits only one at a time to fire. It is the combination of the activation representing syntactic information and that from other sources which determines which constituent wins out over the others and fires. The firing of the winning constituent represents the selection of an item to fill the next output position.

A second problem involves the two types of signals which constituent nodes must send. This problem is handled by having two nodes for each constituent or phrase, one representing the start and the other the end of the unit. The start node signals daughter constituents to be produced, and the end node signals following sister constituents to be produced.

Figure 2 illustrates some of the sequencing information in the \*SEND-MAIL entry. Start-end node pairs are denoted by pairs of small squares surrounded by rectangles with rounded corners. The upper square represents the start, the lower square the end of the word or phrase. Single directional connections are indicated by arrow heads, and pairs of inhibitory connections are denoted by fuzzy lines. The figure includes some sequence connections and the WTA network which represents the competition between the OBJECT-REFERENCE and RECIPIENT-REFERENCE constituents for the position following the VERB. Here the WTA source is the VERB/end node, which sends activation to both the OBJECT-REFERENCE/start and RECIPIENT-REFERENCE/start nodes. These two nodes inhibit each other.

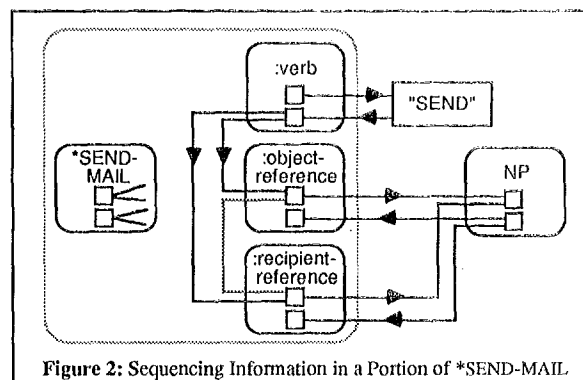


Figure 2: Sequencing Information in a Portion of \*SEND-MAIL

## 5. An Example

### 5.1. Generation

Consider now the generation of sentence (1a): *Instead of calling John, Mary sent him a letter.* Generation begins with the firing of a set of network nodes representing a goal of the speaker. In this case the goal is that the hearer believe that a particular event (the sending of the letter) replaces one previously assumed to occur (the making of a telephone call). This type of goal leads the system to generate a STATEMENT referring to the event preceded by a phrase which denies the assumption (*instead of calling John*). We concentrate here on the generation of the clause beginning with *Mary* and in

particular on the sequencing of the last two constituents.

The event to be referred to is represented as an instance of the general ABSTRACT-TRANSFER predicate (Schank & Abelson 1977) with MARY as the ACTOR, an instance of the concept LETTER as the OBJECT, JOHN as the RECIPIENT, and MAIL as the MEANS of the transfer. We ignore time and tense in order to simplify the discussion. The utterance of the initial *instead of* phrase results in processing of the concepts of MARY and JOHN, so there is residual activation on these nodes and the nodes immediately connected to them. A portion of the network at this point is shown in Figure 3. Nodes with hatched patterns are those with activation below the firing threshold level.

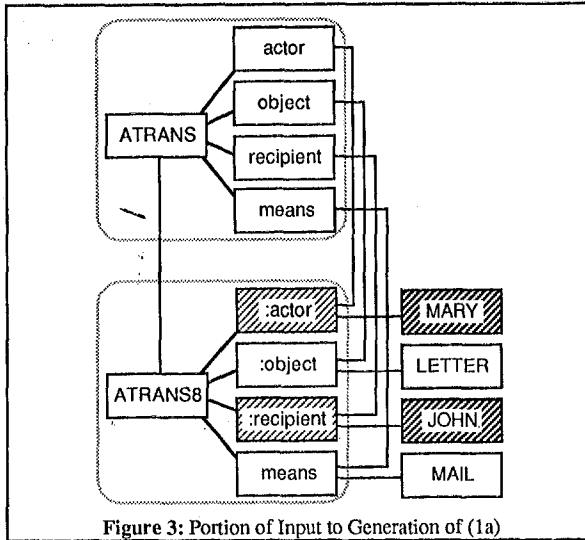


Figure 3: Portion of Input to Generation of (1a)

Activation spreading from ABSTRACT-TRANSFER8 (i.e., the specific transfer instance) converges on a set of verb lexical entries that may be used to describe the input notion. Competition among the CONTENT roles of these entries eventually forces one to win out. For this example, we assume that the \*SEND-MAIL entry would predominate because of the fact that it matches the input MEANS feature, though the entry for the verb *mail* would also be a strong candidate. A simplified view of this lexical entry selection process is shown in Figure 4. The path of activation spread is indicated by arrows in the figure, blackened nodes are those that fire initially, and nodes with wide borders are those that fire in response to the spread of activation. The fuzzy lines emanating from \*SEND-MAIL:CONTENT are inhibitory connections to other verb CONTENT roles.

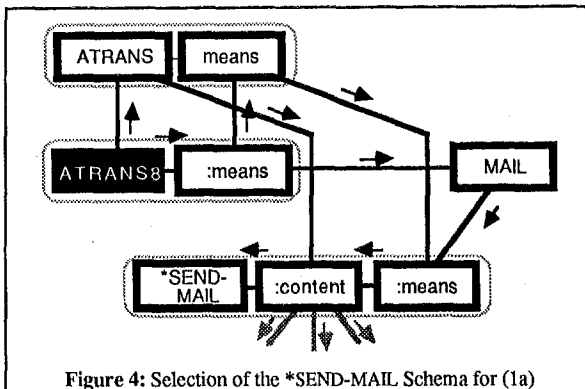


Figure 4: Selection of the \*SEND-MAIL Schema for (1a)

At the same time, activation spreading from ABSTRACT-TRANSFER8 causes the primed RECIPIENT node to fire, leading to a series of firing nodes and eventually to the priming of the RECIPIENT-REFERENCE role in the \*SEND-MAIL entry. This process is shown in Figure 5.

Once the \*SEND-MAIL entry has been selected, activation

spreads through it, resulting in the priming of the nodes representing the constituents of the clause. At the same time activation has also spread to the constituent nodes of the higher-level CLAUSE schema. The connections within this schema determine the order of the SUBJECT and VERB in the sentence. The fact that the event referred to occurred before the time of speaking also leads to the selection of the PAST-CLAUSE schema, and this in combination with the \*SEND-MAIL schema results in the firing of the node representing the word *sent*. For the purposes of this paper, we ignore the details of these processes.

When the verb has been produced, the VERB/end node in the \*SEND-MAIL entry fires. From here activation spreads to the nodes representing the beginnings of the two possible following constituents: RECIPIENT-REFERENCE/start and OBJECT-REFERENCE/start. These nodes compete with one another via a WTA network. In this case the priming on the RECIPIENT-REFERENCE/start node leads this constituent to win out over OBJECT-REFERENCE/start. The situation at this point is shown in Figure 6.

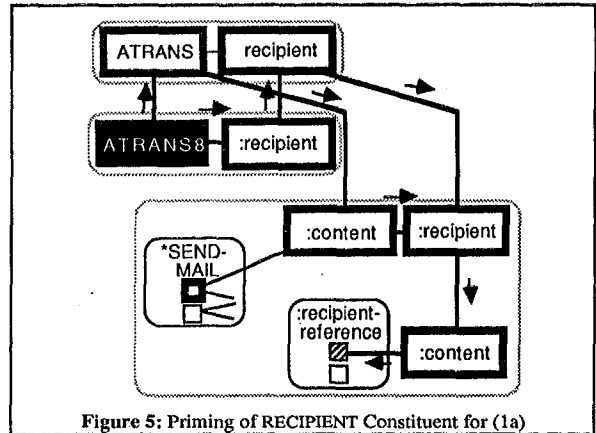


Figure 5: Priming of RECIPIENT Constituent for (1a)

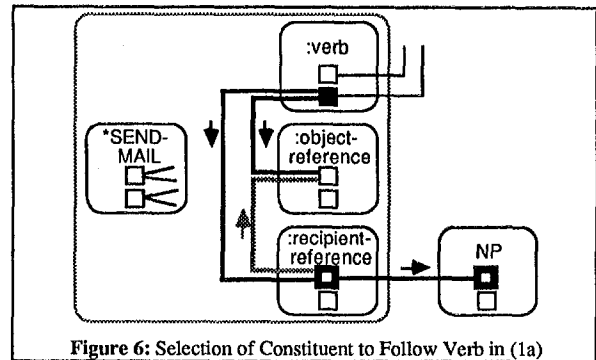


Figure 6: Selection of Constituent to Follow Verb in (1a)

Next the NP schema takes over. At this point there is competition between the schema for pronouns and that for full NPs. The pronoun schema wins out when there is evidence that the hearer is currently conscious of the referent. In this case such evidence is available in the form of residual activation resulting from the reference to John in the phrase *instead of calling John*. For details on how spreading activation and competition implement the selection of pronouns over full NPs, see Gasser (1988).

When the NP is complete, activation is sent back to the RECIPIENT-REFERENCE/end node, which then activates the nodes representing the two possibilities for what follows. One is that the clause is complete. This option would be the appropriate one if the RECIPIENT-REFERENCE had followed the OBJECT-REFERENCE (as in *Mary sent a letter to John*). The other option, the one that is appropriate for this example, is that the OBJECT-REFERENCE follows. The reason that both possibilities need to be represented is that the system has no explicit memory for what has or has not already been generated. The weights on the two connections are such that the second alternative is the default and will be preferred in this case. That

is, OBJECT-REFERENCE wins out, and the OBJECT-REFERENCE/start node fires. As shown in Figure 7, the selection of the OBJECT-REFERENCE role leads eventually to the firing of the OBJECT role in ATRANS8 and the LETTER node.

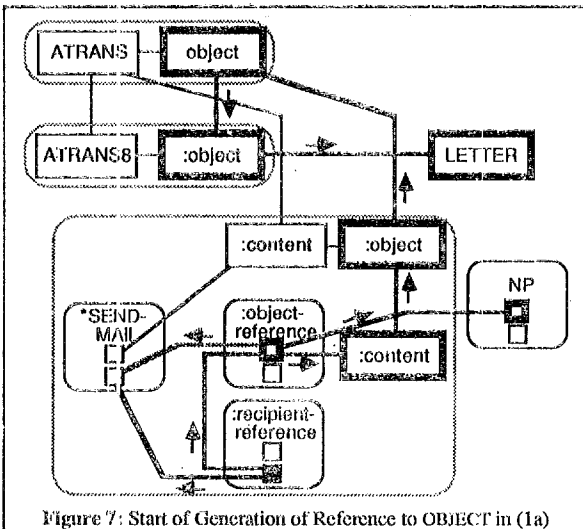


Figure 7: Start of Generation of Reference to OBJECT in (1a)

Again control is passed to the NP schema. Here two further selections take place. The fact that there is no evidence that the hearer knows the referent leads to selection of the INDEFINITE-NP schema over the DEFINITE-NP schema by default. INDEFINITE-NP specifies the indefinite article *a*. Finally, the lexical entry \*LETTER is selected as a result of activation spreading from the LETTER node. This schema provides the noun *letter* for the OBJECT NP.

Once the final constituent is complete, activation is sent back to the OBJECT-REFERENCE/end node. Again there are two possibilities for what may follow, the end of the clause or the *to* case marker and the following RECIPIENT-REFERENCE constituent. Note, however, that there is an inhibitory connection from RECIPIENT-REFERENCE/end to RECIPIENT-MARKER/start. That is, the completion of the RECIPIENT-REFERENCE effectively prevents the later generation of the case marker, and as a consequence the repetition of the RECIPIENT-REFERENCE itself. The state of the network at this point is shown in Figure 8. The fuzzy filled pattern on RECIPIENT-MARKER/start indicates that the node is inhibited.

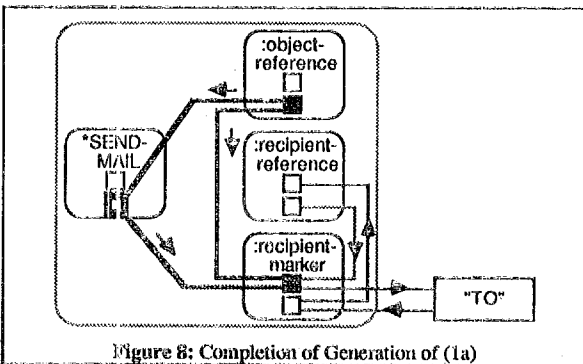


Figure 8: Completion of Generation of (1a)

In this example we have made use of sequencing information found in the lexical entry \*SEND-MAIL. This sort of information also appears in more general lexical entries such as \*SEND and in non-lexical GUs such as ATRANS-CLAUSE, the schema for clauses referring to an ABSTRACT-TRANSFER. If a specific entry lacks the required information, a more general schema is used automatically.

## 5.2. Parsing

Now consider how the same information would be used in the parsing of the sentence *Mary sent him a letter*. Recognition of the word *Mary* leads to the selection of the \*MARY entry and the

consequent firing of the MARY node. Recognition of the word *sent* results in the selection of the \*SEND entry, which is similar to the more specific \*SEND-MAIL entry shown in Figures 1, 2, 4, 5, 6, 7, and 8. Activation is sent immediately to the SUBJECT constituent of the entry, resulting eventually in the firing of the ACTOR node. It is the close proximity of the firing of MARY and ACTOR which represents the role binding aspect of parsing. Recall from 3.3 above that there is currently no way to record this binding permanently in the system's memory.

The firing of the VERB/end node in the \*SEND schema leads, as in the generation of the same sentence, to the activation of nodes for both of the constituents which may follow. At this point neither of these constituents has enough activation to fire. The activation that is present represents the expectation that there will now be a reference to either the RECIPIENT or the OBJECT.

Next the word *him* is recognized, leading to the activation of all male humans that the system is currently "thinking about". There is only one such entity, John, and the JOHN node then fires. Activation spreads to nodes for features of John including the HUMAN node. Since humanness is a default property of the RECIPIENT of an ABSTRACT-TRANSFER, this last node is connected to the RECIPIENT node, which can now fire, sending activation in turn eventually to the RECIPIENT-REFERENCE/start node in the \*SEND schema. The additional activation now causes this node to fire, representing the system's recognition that the current constituent refers to the RECIPIENT rather than the OBJECT of the ABSTRACT-TRANSFER.

From this point on, the process, at least with respect to sequencing, is similar to what goes on during generation. After the appearance of the word *him*, activation spreads from the RECIPIENT-REFERENCE/end node to the nodes representing the two possible alternatives, the end of the clause or the appearance of the OBJECT-REFERENCE. The latter will predominate in this example once the beginning of the NP *the letter* is recognized. Following the completion of this NP, there will again be two alternatives. In this case the CLAUSE/end option will win out, as in the generation case, because of inhibition on the node for the alternative.

## 6. Implementation and Coverage

The model described in this paper is implemented in a program called CHIE. The program has two components, a hand-coded memory network representing both world knowledge and linguistic knowledge and a set of procedures implementing spreading activation and inhibition through WTA networks. CHIE generates sentences in English and Japanese given input in the form of activated network nodes representing speaker goals. The model has been tested for a small fragment of the grammars of these languages: simple declarative and interrogative clauses and noun phrases with adjective modifiers. In addition to obligatory constituents like those in the example above, the program handles optional and optionally iterating constituents. The program also "parses" the structures that it generates using the same memory and the same basic procedures, but, as noted in 3.3, it does not save a semantic interpretation; that is, it does not know how to create schema instantiations with role bindings. (See Dolan & Dyer 1987 for an approach to this problem within the connectionist framework.)

## 7. Related Work

While not adhering strictly to any familiar theoretical framework, the present model has aspects in common with the Phrasal Lexicon approach, (e.g., Jacobs 1985, Zernik & Dyer forthcoming), with phrase-oriented work in linguistics and psycholinguistics (e.g., Fillmore, Kay, & O'Connor 1986, Pawley & Syder 1983, Peters 1983), with other localized connectionist models (e.g., Cottrell & Small 1983, Waltz & Pollack 1985), and with psychological models making use of spreading activation (e.g., Dell 1986, MacKay 1987).

The approach described in this paper is apparently the first effort to model language generation totally within the connectionist framework. There have been more limited efforts, however. Kukich (1986) has looked at the distributed representation of phrases and how these might be learned; however, she does not consider interacting

factors in sequencing. Dell (1986) has developed a psycholinguistic model using spreading activation for selecting candidate items, but his model deals mainly with effects at the level of morphology and phonology. Hasida, Ishizaki, and Isahara (1986) use a spreading activation mechanism to select important information for generating abstracts. We view these three areas of research as complementing our model.

Unlike distributed connectionist models, e.g., those described in McClelland, Rumelhart, and the PDP Research Group (1986), memory in the present model is localized; that is, each concept is represented by a single memory node. This mode of representation brings with it certain disadvantages, in particular, the property that processing does not degrade gracefully when a portion of the memory is destroyed. On the other hand, the model maintains the constituency that is basic to symbolic models and the need for which, as Fodor and Pylyshyn (1988) argue, presents the most serious challenge to distributed models. It should be clear from this paper that constituency is fundamental to the way in which sequencing information is represented and used in the model.

Within connectionist models the approach to sequencing adopted here is most similar to that suggested by Feldman and Ballard (1982) in that sequencing relations are represented explicitly in the form of connections. What we have done is elaborated on this approach to deal with the complications that arise in the generation of language, in particular, the interaction of semantic and syntactic effects in sequencing. In addition, our model appears to be the first connectionist model to make use of the same representation of sequencing information for generation and parsing (but see MacKay 1987 for a psychological theory with similar claims).

Our work can also be contrasted with other approaches to syntax in language analysis. In some respects the flow of activation through entries such as \*SEND and \*SEND-MAIL resembles what goes on in recursive transition networks; however, there are three important differences. First, in this model syntax and semantics interact in processing, and the output of the system when used in parsing represents both a syntactic and a semantic analysis of the input. Second, the network can be used in both the generation and the parsing directions. Third, the formalism permits the representation and use of tendencies as well as absolute constraints regarding sequencing.

## 8. Conclusions and Future Work

In this paper we have argued that since sequencing in language involves competition among various quantitative factors, it can be profitably modelled within a connectionist framework, and we have presented a localized connectionist scheme for representing and using sequencing information in language processing. Key features of the approach are the representation of phrasal units and their constituents as pairs of network nodes, one for the start and one for the end of the sequence; the representation of ordering constraints and tendencies as weighted connections; and the use of winner-take-all networks to impose sequentiality on a parallel spreading activation mechanism.

The model has been tested for a small set of simple clause and NP types. It remains to be seen whether it can cover the range of sequencing constraints and tendencies found in human languages, for example, the requirement in German and Dutch that the verb appear in second position in clauses and the apparent total lack of syntactic ordering conventions in some Australian languages. We are currently attempting to extend the model to handle such features. We are also working on a means of incorporating backtracking (in both generation and parsing) into the model to simulate garden path effects.

A further area of future research is the incorporation of a learning capability in the model. The major weakness of the model thus far is the need to hand-wire the memory network, in particular to set the weights on the connections. What we are working toward is a model that is able to adjust its own connection weights in response to presentations of input-output mappings, as is done in many distributed connectionist approaches.

## Notes

<sup>1</sup>The research reported on here was supported in part by grants from

the ITA Foundation and the JTF program of the U.S. Department of Defense.

<sup>2</sup>Address from August 15, 1988: Computer Science Department, Indiana University, Bloomington, Indiana 47405, USA

<sup>3</sup>In its present form the entry applies to active clauses only. For simplification we have ignored the possibility of passives.

## References

- Bock, J. K. (1982) 'Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation.' *Psychological Review* 89, pp. 1-47.
- Cottrell, G. W. & S. L. Small (1983) 'A connectionist scheme for modelling word sense disambiguation.' *Cognition and Brain Theory* 6, pp. 89-120.
- Dell, G. S. (1986) 'A spreading-activation theory of retrieval in sentence production.' *Psychological Review* 93, pp. 283-321.
- Dolan, C. P. & M. G. Dyer (1987) 'Symbolic schemata, role binding, and the evolution of structure in connectionist memories. Proceedings of the IEEE First Annual International Conference on Neural Networks.
- Fahlman, S. E. (1979) *NETL: A System for Representing and Using Real-World Knowledge*. Cambridge, MA: MIT Press.
- Feldman, J. A. & D. H. Ballard (1982) 'Connectionist models and their properties.' *Cognitive Science* 6, pp. 205-254.
- Fillmore, C. G., P. Kay, & M. C. O'Connor (1986) 'Regularity and idiomaticity in grammatical constructions: The case of *let alone*.' Unpublished manuscript.
- Fodor, J. A. & Z. W. Pylyshyn (1988) 'Connectionism and cognitive architecture: A critical analysis.' *Cognition* 28, pp. 3-71.
- Gasser, M. (1988) *A Connectionist Model of Sentence Generation in a First and Second Language*. Unpublished doctoral dissertation, University of California, Los Angeles.
- Hasida, K., S. Ishizaki, & H. Isahara (1986) 'A connectionist approach to the generation of abstracts.' In Kempen, G. (ed.), *Natural Language Generation*. Dordrecht: Martinus Nijhoff, pp. 149-156.
- Jacobs, P. S. (1985) 'PHRED: A generator for natural language interfaces.' *Computational Linguistics* 11, pp. 219-242.
- Kukich, K. (1986). 'Where do phrases come from: Some preliminary experiments in connectionist phrase generation.' In Kempen, G. (ed.), *Natural Language Generation*. Dordrecht: Martinus Nijhoff, pp. 405-421.
- Langacker, R. W. (1987) *Foundations of Cognitive Grammar* (Vol. 1). Stanford, CA: Stanford University Press.
- MacKay, D. G. (1987). *The Organization of Perception and Action: A Theory for Language and Other Skills*. New York: Springer-Verlag.
- McClelland, J. L. & A. H. Kawamoto (1986) 'Mechanisms of sentence processing: Assigning roles to constituents of sentences.' In McClelland, Rumelhart, & the PDP Research Group (eds.), pp. 272-325.
- McClelland, J. L., D. E. Rumelhart, & the PDP Research Group (eds.) (1986) *Parallel Distributed Processing. Explorations in the Microstructures of Cognition: Vol. 2: Psychological and Biological Models*. Cambridge, MA: MIT Press.
- Pawley, A. & F. H. Syder (1983) 'Two puzzles for linguistic theory: Nativelike selection and nativelike fluency.' In Richards, J. C. & R. W. Schmidt (eds.), *Language and Communication*. London: Longman.
- Peters, A. M. (1983) *The Units of Language Acquisition*. Cambridge: Cambridge University Press.
- Schank & Abelson (1977) *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Erlbaum.
- Waltz, D. L. & J. B. Pollack (1985) 'Massively parallel parsing: A strongly interactive model of natural language interpretation.' *Cognitive Science* 9, pp. 51-74.
- Zernik, U. & M. G. Dyer (forthcoming) 'The self-entending phrasal lexicon.' *Computational Linguistics*.