Chapter 12
A Connectionist Approach to
Generation of Simple Sentences and
Word Choice

Jugal Kalita
Department of Computer Science
University of Colorado

Lokendra Shastri
Department of Computer and Information Science
University of Pennsylvania

This chapter discusses the design and implementation of a connectionist system for
geneneration of simple, well-formed English sentences. The design employs several
levels of interacting units for making appropriate decisions. It uses a simple tech-
nique for specifying assignment of input concepts to roles in a sentence and also
has a reusable subnetwork for the expansion of noun phrases. The same NP-
subnetwork is used for the expansion of noun phrases corresponding to various
conceptual roles of the generated sentences. The input to the system consists of
parallel activation of a cluster of nodes representing a conceptual specification of
the sentence, whereas the output is in the form of sequential activation of nodes
corresponding to the words constituting the sentence. The system can produce
simple sentences in both active and passive voices, and in several tenses. We also
discuss how the system can be augmented to account for choice of lexical items
based on simple pragmatic criteria. Results of simulation experiments performed
are included.
1. INTRODUCTION

It is generally accepted (Webber, 1984; McKeown, 1985) that text generation involves three distinct sequential phases—content determination, text planning, and surface generation. Content determination involves identifying information that needs to be included in the generated text. Text planning is concerned with appropriately sequencing the intended contents in order to achieve coherence in the generated text. The final phase—surface generation—produces the actual sentences given their internal representations. In the system discussed in this chapter, we are not involved with the first two phases. Our emphasis is on the generation of well-formed sentences, assuming that the initial two phases have been successfully performed.

There are several approaches to surface generation. These include the approach based on functional grammar as used by McKeown (1985) in her TEXT system, the propositional logic based system employed by Appelt (1983), the knowledge-driven approach taken by Jacobs (1985), and the stepwise refinement approach taken by McDonald (1983) in the MUMBLE system. The functional grammar approach is nondeterministic; the generation of a sentence from a conceptual specification involves the process of unification, which is slow and inefficient, taking potentially exponential computation time. Appelt assumes homogeneity of various decision processes; his approach, which requires implementation of theorem proving techniques, is also not easily amenable to parallelism. McDonald employs several levels of processing to achieve the system’s goal. Consequently, processing requirements are relatively simple at each level, enhancing efficiency and modifiability. It is deterministic in the nature of processing involved.

Our approach to surface language generation is similar to that of McDonald. We have modeled the generation process as a hierarchy of processing levels. Decisions are made at each of these levels until nodes corresponding to the words constituting the sentence are activated in an appropriate order. We implement the various processing levels by using the connectionist model of computation in the spirit of Feldman and Ballard (1982). The techniques employed in our implementation have been influenced by the systems for word-sense disambiguation and parsing reported by Cottrell (1985) and by Waltz and Pollack (1985). Our system successfully generates simple English sentences, given a nonlinguistic specification of their contents. Recently, Ward (1988) has discussed a very general outline of a similar system, which operates on the principle of energy flow through a semantic network (from input concepts to output words), although no specifics have been presented. Our approach also bears certain similarities to the knowledge-intensive approach taken by Jacobs (1985), which, although hierarchical, is not a connectionist formulation, and does not have clearly demarcated levels of abstraction. Gasser’s (1988; Gasser & Dyer, 1988) approach is also modeled partially after Jacobs’s (1985) ACE-based sentence generator.
In Section 2 of this chapter, we introduce the various levels of processing into which our network is partitioned, and the relations among these levels. Section 3 describes the structure and the behavior of the units used in building the network. Sections 4–6 discuss details of various parts of the network—including the reusable NP-subnetwork input specification and winner-take-all networks. Finally, Section 7 gives results of a simulation performed showing the manner in which nodes in the output sentence are activated. Section 8 discusses how the approach can be extended to incorporate decisions regarding simple lexical choice based on pragmatic criteria. The research reported in this chapter was discussed briefly in Kalita and Shastri (1987).

2. THE LEVELS OF PROCESSING

In our approach, there are several levels of processing in the generation of a well-formed sentence from its conceptual specification in our generational paradigm. In this respect, our chosen approach is similar to the approaches to perceptual processing reported by McClelland and Rumelhart (1981) and Sabbah (1985), where each level of processing is concerned with forming a representation of the input at a different level of abstraction. In our system, levels represent various decision steps that need to be taken in order to generate a sentence.

The main levels of processing employed in the system are

- input level
- realization-class level
- choice level
- constituent level
- morphology level

The relationships among the various levels are shown in Figure 12.1. The type of processing performed at each level is discussed below.

Each level of processing is carried out by one or more specially designed cluster of nodes. The input level nodes represent a communicative goal (along with some constraints to be discussed later). This information is in the form of a nonlinguistic conceptual specification of what needs to be conveyed through an intended utterance. The input level nodes represent various concepts, both actions or objects. Examples of input level nodes representing concepts are concept-eat, concept-monkey-1, concept-banana-1, etc. We may also refer to such input level nodes as concept nodes or concept units. A concept is an abstract entity, whereas a concept node (or a concept unit) is the representation of the concept in the network under consideration. We use the terms node and unit interchangeably in this chapter.

Each concept node is linked to a single realization-class node. That is, several
concept nodes may be linked to the same realization-class node, but each concept node is connected to a unique realization-class node. The realization-class node, which is connected to a concept node, determines how that concept is translated into English. The translation of all concepts whose corresponding nodes are connected to a realization-class node proceeds in an identical fashion. Realization classes partition the concept space into several groups. Examples of realization-class nodes are svo and individual-item. The svo node requires concept nodes linked to it to be translated into a sentence where the subject, verb and the object are all present in this particular order. Concept units corresponding to transitive verbs such as eat, give, etc., are connected to the svo node. Concept nodes such as concept-monkey-1, concept-baboon-1, concept-banana-1, which denote individual objects that can be expressed in English in terms of noun phrases, are each connected by an excitatory link to the individual-item node in the realization-class level.

Each realization-class node has activation links to one or more choice level nodes. The choice nodes connected to a realization-class node form a winner-take-all network (Feldman & Ballard, 1982; Shastri, 1985). In other words, only one of them can be active at any instant of time. Each choice node connected to a

---

1 The weights on all links discussed in this and following sections are 1 unless noted otherwise.
realization-class node represents a different grammatical way of realizing a concept linked to that realization-class node. For example, the realization-class node svo is connected to two choice nodes: active-svo-choice and passive-svo-choice. This allows a concept such as concept-eat (whose corresponding realization-class node is connected to the svo node) to be translated into either an active or a passive sentence after its subject and object roles are filled.

The choice nodes have other activation links from constraint-specification nodes (which are also input nodes) incident upon them. Activation on each input link is necessary for a choice node to become active. The constant-specification nodes must be activated by the text planner before the process of surface generation is started. Examples of constraint-specification nodes include current-voice-is-active, current-aspect-is-perfect, etc. In order for the active-svo-choice node to be active, it must receive activation inputs both from the svo node (a realization-class node) and the current-voice-is-active node (a constraint-specification node).

A choice node has activation links to one or more constituent level nodes. The constituent nodes specify the roles that need to be filled in order to achieve a particular choice of realization of an input concept. Examples of constituent nodes connected to the choice node active-svo-choice are subject-slot, active-svo-verb and object-slot nodes. The activation of the constituent nodes is sequenced. Thus, the translation of a concept node connected to the svo node (a realization-class node), assuming the current-voice-is-active node is active, involves filling in three roles in sequence: subject, verb and object. Sequencing is accomplished using sequencer nodes, discussed later.

Each constituent node is either connected to one or more nodes that handle morphology or to a cluster of nodes that specify role association. Activation of a constituent node may result in excitation of one or more word nodes in a predetermined sequence. If a constituent is connected to a node that specifies role assignment, its activation starts expansion of another realization-class node corresponding to the conceptual specification of a part of the sentence (e.g., the subject or object role of the sentence).

The morphology handling level contains two types of nodes: generic-word and word nodes. It is assumed that activation of a word node leads to the word being spoken aloud or written out. A generic-word node represents a word whose lexicalization is dependent upon the grammatical or pragmatic context. For example, the node generic-word-eat is connected to the word nodes word-eat and word-eats. It should be noted that complex forms of verbs such as has been eating (i.e., three word nodes: word-has, word-been and word-eating whose activation is sequenced) are also connected to the generic-word-eat node. Depending on constraints such as current tense, current voice and current aspect, one of these sets of word nodes is activated.

In addition to activation links from constituent-level and constraint-specification nodes, the morphology nodes may have activation links from
concept nodes via the cluster of nodes specifying role association. The realization-class node to which a concept node is connected determines the structure of the phrase or clause that results from the concept's translation through its links to choice and constituent nodes. One or more of the word nodes that need to be activated to fill in the slots in the translated phrase or clause may be determined directly by the concept node itself by having activation links onto specific generic-word or word nodes.

3. UNIT TYPES AND UNIT BEHAVIOR

Each of the levels of processing discussed in the previous section has one or more unit types associated with it. A unit type refers to all units with the same general structure, functional capabilities, and used for similar purposes, most often in the same level of computation. Thus, we have the following unit types: concept, constraint-specification, realization-class, choice, constituent, generic-word, and word.

Input nodes, which include concept and constraint-specification nodes, are activated by the text planner. They are activated simultaneously, and thereafter they drive the process of generation.

Generation of a well-formed sentence involves activating the nodes that correspond to words in the sentence. However, parallel activation of word nodes cannot be considered as generating a sentence. Sequencing of activation of word nodes is imperative in order to generate a meaningful sentence. The necessity for a sequencing circuit arises from the manner in which the input is fed and the sequential manner in which the output is required to be produced. The sequencing technique discussed next is similar to the one employed by Cottrell (1985). The sequencing mechanism employed by Gasser (1988; Gasser & Dyer, 1988) is similar in principle, although he uses two separate nodes—start node and end node—with each constituent instead of using two special-purpose sites in the units.

In order to achieve appropriate word sequencing, a new unit type has been defined: the sequencer type. In particular, sequencer nodes also assist in meeting the crucial requirement that only one word node remain active at any instant of time, since only one word can be written or pronounced at one time for a meaningful sentence to be generated. Feedback from a corresponding sequencer node turns off a word node a few cycles after it has been activated. Sequencer nodes also facilitate the deactivation of units that have already participated in the generation process and are no longer required.

Each unit in the network has several sites. The nature of these sites is dependent on the unit type. Among the sites are: or, and, expansion-completed, and reuse-site. An or site computes it value to be the highest among all its weighted input signals, whereas the computation carried out at the and sites
involves selecting the minimum of its weighted inputs as its value. The function of the other two sites is discussed later.

The units can be in three possible states. They are initial, active, and inert. To start with, all nodes are in the initial state. A node is switched to the active state on receiving appropriate excitation signals at its or or and sites. Once in the activation state, a unit remains so until such time when a signal is received at its expansion-completed site from a sequencer unit; this forces a unit to inert state. Thus, when all activity ceases, the nodes which participated in the processing are left in inert state.

In our earlier implementation (Kalita & Shastri, 1987), unit behavior was in the form of step functions. Receipt of any positive activation, irrespective of its magnitude, used to drive the unit potential to high level at which it used to be stable until a positive signal (again, irrespective of its magnitude) was received at the expansion-completed site. Currently, we use a potential function which is similar to the one used by McClelland and Rumelhart (1981) and Shastri (1985). As a result of this, the potential curves obtained now have smoother form compared to the step functions we had earlier, demonstrating how the potential of a unit to be pronounced rises gradually due to influence of various factors. In other words, decisions are made not hastily, but by gradual collection of evidence from various sources of support and inhibition. The function used now is

\[ p(t + 1) = \delta \times p(t) \text{ in inactive state} \]

\[ p(t + 1) = \delta \times p(t) + (1 - \delta) \times \text{input} \text{ in active state} \]

where

\[ \text{input} = \sum_{\text{all-sites}} \text{SiteFunction(inputs at the site)} \]

\( \delta \) is the potential decay rate. If there is no external input to a unit at time \( t \), its potential is reduced by a factor of \( \delta \) at the next time point \( t + 1 \). When there are external inputs to a unit at time \( t \), its potential at time \( t + 1 \) is determined by its potential at time \( t \) as well as the external inputs at time \( t \). The value of the cumulative external inputs is multiplied by a factor of \( 1 - \delta \) to obtain the contribution of the external inputs to the potential of the unit at time \( t + 1 \).

However, we ran into unexpected problems using the site function. In some cases, although one unit was clearly the winner, its potential after all inputs had stabilized was quite low as an absolute magnitude. In other words, although the winner prevailed over its competitors and won, its strength or potential was exhausted in the process to a considerable degree. As a result, it could not drive other units which it had to drive since the input at the other units did not go beyond the minimum stipulated threshold. Hence, we modified the potential function in active state to

\[ p(t + 1) = \delta \times p(t) + \beta \times (1 - \delta) \times \text{input} \]
where

\[
\beta = \begin{cases} 
1 & \text{if } p(t) \leq \text{HYPER\_THRESHOLD} \times \text{MAX\_POTENTIAL} \\
\frac{\text{MAX\_POTENTIAL}}{\text{current\_potential}} & \text{otherwise}
\end{cases}
\]

\text{MAX\_POTENTIAL} is a number beyond which a unit's potential is not allowed to rise. In our implementation it is 10. \text{HYPER\_THRESHOLD} is a fraction such that if the unit's potential crosses \text{MAX\_POTENTIAL} \times \text{HYPER\_THRESHOLD} it is assumed to be hyperactive after which its potential rises more rapidly.

4. REUSABLE SUBNETWORKS

The unit behavior discussed thus far is sufficient to generate simple sentences, but in order to achieve better resource utilization, it is necessary to introduce another site called the \textit{reuse-site}. A positive input at this site of a unit in \textit{inert} state causes the unit to change its state to \textit{initial} state so that the unit can be reused. Reuse-sites are used only for the nodes in the subnetwork for noun phrase expansion.\footnote{Although \textit{reuse-sites} are present in all units, they remain unutilized at the current time in units outside the NP-subnetwork. This is because, among all the syntactic constituents, only noun phrases may be used more than once in the types of sentences that we consider. We may use the \textit{reuse-site} in other units in future, when the syntactic coverage of the system is broadened.} This enables us to use the same noun phrase expansion subnetwork for expanding the subject as well as the object of a sentence, resulting in efficient resource utilization. Using two separate subnetworks for noun phrase expansion—one for the subject and the other for the object—would lead to a wastage of resources. Since the task is essentially the same, we have decided to design a reusable \textit{NP-subnetwork}. When a noun phrase expansion is complete, a specially designated unit in the NP expansion subnetwork gets activated. This unit, then, sends a positive activation to all units in the noun phrase subnetwork at the \textit{reuse-site}, as a result of which the units become ready to resume their activity.

This, in our view, is a significant feature of our approach. Supposing we are generating an active sentence; the subject NP needs to be expanded first. In order to achieve this, the subject-slot constituent node is activated. This initiates the expansion of the subject NP. Once this expansion is complete, all nodes that participated in this expansion in the NP subnetwork are reset by a signal at the \textit{reuse-site} as explained earlier and are available for reuse. After subject NP expansion, the active verb phrase is generated. Following this, the same NP expansion subnetwork is used to generate the noun phrase corresponding to the object.

In order to generate active as well as passive sentences, we have a \textit{passive-svo} choice unit connected to the realization class unit \textit{svo}. The passive choice
unit gets activated only when it receives activation from the realization class unit \textit{svo} and the constraint specification unit \textit{current-voice-is-passive}. The sub-network employed for generation of subject and object noun phrases for active sentences is also used for passive sentences without any modification. New sequencing nodes have been introduced so that the expansion of the object phrase precedes that of the verb and agent phrases for the passive case. Units for handling passive forms of verbs have also been implemented.

The distinction between the two inactive states discussed earlier (initial and inert) is introduced to prevent oscillation of a node between active and inactive states. This requirement is extremely important, in particular for word nodes, in order to achieve proper sequentialization of utterance. In our implementation, it was required of a unit to remain in inactive state once there is a transition from active to inactive state, to prevent unexpected potential and state oscillations. A node which is inactive to start with plays its appropriate role in the process of generation by becoming active on receiving excitatory inputs. Consequently, it activates other units, and in turn, it becomes inactive again (inert). However, the arrival of a positive input at the \textit{reuse-site} causes a unit in inert state to make a transition to initial state, enabling its participation in repetitive processing.

Nakagawa and Mori (1988) present a parsing mechanism that uses copies of parse subnetworks in order to process parts of a sentence belonging to the same syntactic category. Their subnetwork-copying mechanism is patterned after the \textit{connection information distribution} (CID) mechanism proposed by McClelland (1986). It may be worthwhile to investigate possibilities of using a similar approach for a more advanced surface generator, although for our current goals, our technique is quite satisfactory.

5. BINDING CONCEPTS TO ROLES IN INPUT SPECIFICATION

The mechanism by which the input to the network is specified is described with the aid of Figure 12.2. We associate concepts with conceptual roles such as subject and object in a sentence. Corresponding to each object that can fill in the subject/object role of a sentence, we have a concept node such as \textit{concept-monkey-1}, \textit{concept-banana-1}, etc as discussed earlier. Each of these concept nodes is connected to the realization-class unit \textit{individual-item} (this is shown in Figure 12.3). Each of these concept units is also connected to two binder units for the purpose of role association. Consider the concept unit \textit{concept-monkey-1}. It is connected to two units labeled \textit{e-s-1} and \textit{e-o-1} in the diagram. \textit{e-s-1} is called a subject-binder unit; \textit{e-o-1} is an object-binder unit. There are two such units for each concept that can fill the role of subject/object. In this example, \textit{e-s-1} plays a role in associating \textit{concept-monkey-1} unit to the subject role; and \textit{e-o-1} helps associate \textit{concept-monkey-1} to the object role. It should
be noted that each subject-binder node (for each concept) has an activation link from the subject-slot unit (which is a constituent unit driven by the svo-active/svo-passive choice units for the realization-class unit svo). Similarly, the object-binder node for each concept has an activation link from the object-slot constituent unit.

Several other units are also used in this process. Two such units are the global-subject-enable and global-object-enable nodes. These nodes are always kept active. There is a link from global-subject-enable node to the subject-binder node for each concept node. There is a similar link from global-object-enable to the object-binder node for that concept. Initially, each such link is in a disabled state. Appropriate links are enabled before processing a sentence to specify role assignments. Enabling of role assignment links is done at input time before processing is initiated. This is required in order to correctly specify at a conceptual level the correspondence among concepts and roles in the sentence to be generated. If, however, our system needs to be interfaced with higher level generation systems that perform text planning, such enabling of links will have to be performed automatically. This is a problem which future research must address.
In order to assign the concept-monkey-1 unit to the subject role, the link from the global-subject-enable to the subject-binder unit for concept-monkey-1 (here, unit e-s-1) is enabled at input time. (Note that all other links from global-subject-enable unit to all other subject-binder nodes are still disabled.) And, similarly, to assign another unit (say, concept-banana-1) to the object role, the link from global-object-enable unit to the object-binder unit for concept-banana-1 is also enabled. So, the crucial step in specifying the input properly constitutes enabling the linkages appropriately before the processing of a sentence starts.

The subject-binder and the object-binder nodes are such that they need all their inputs to get activated. A subject-binder node provides excitatory input to the subject-driver node. Excitation from any subject-binder node excites the subject-driver node. Similarly, there is an object-driver node which receives excitatory input from the object-binder nodes. The subject-driver and the object-driver nodes have excitatory inputs to an np-expansion-driver node. This node drives a node called the binding-completion-signal, which initiates the expansion of the choice nodes connected to the individual-
item realization-class node. Thus, excitation of the binding-completion-
signaler unit, which sits between the realization-class node individual-item
and its choice units, initiates the excitation of units that leads to the expansion of
the noun phrase corresponding to the subject or object of a sentence.

The choice unit active-svo has excitatory inputs to subject-slot, active-
svo-verb, and object-slot units—the activation of these units is sequenced.
Thus, during the expansion of the subject, only the subject-slot unit is active;
as a result, the subject-driver node is activated (object-driver node is off).
This leads to the generation of a noun phrase corresponding to the subject of the
sentence. Similarly, when the object-slot unit is active, the object-binder unit
corresponding to the object concept gets activated. This activates the object-
driver node which finally results in the generation of a noun phrase correspond-
ing to the object concept.

6. IMPLEMENTING WINNER-TAKE-ALL NETWORKS

Winner-take-all networks are required for all choice-level clusters (Kalita &
Shastri, 1987). The manner in which we have implemented this is similar to
(Shastri, 1985) and is discussed below in brief.

We have a cluster of competing units one of which is going to win, a
maximum-calculator unit, and a competition-status unit. Each compet-
ing unit has two specialized sites: max-site and a booster-site. The max-site gets
inputs from the corresponding maximum calculator unit; this unit computes the
highest value among all the computing units. Thus, at the max-site of a comput-
ing unit we have the maximum value among all the competitors at the last instant
of time. That is, the information is delayed by a unit of time. Although delayed,
it is helpful in providing information to a computing unit regarding how it is
performing in the competition—whether it is losing or winning, etc.

Booster-site is used for solving the problem mentioned earlier. It gets inputs
from the corresponding competition status unit. When competition is in progress
among two or more units, the maximum-calculator unit computes the maxi-
mum of all input values. The values input to it are the potential values of the
competing units. The potential function used for competing units in a winner-
take-all network is

\[ p(t + 1) = \delta \times p(t) + \frac{\text{MAX\_POTENTIAL}}{\text{current\_potential}} \times (1 - \delta) \times (\text{input} + p(t)) - \text{maximum among all competing values} \]

The competition status unit determines if competition is over or still in progress.
If all but one of the competing units is active, or if the potential of all but one of
the competing units is quite low, the competition status unit's potential goes high.
If the ratio between the maximum value and the second maximum value is greater than or equal to a constant $\alpha$, competition is assumed to be over, since we assume that the losing units will not be able to recover. The value of $\alpha$ currently used is 10. This value has been chosen after considerable experimentation. If this signal is high, the competing unit compares its own potential with the maximum potential and sees if it itself is the maximum. If it is not the maximal unit, it reduces its potential slowly to zero. If it is the maximal unit, then it increases its potential towards the constant MAX_POTENTIAL, using the potential function discussed earlier.

7. AN EXAMPLE SIMULATION

We now briefly run through an example simulation showing the various steps involved in generating a sentence given the conceptual specification of its contents. The network for the simulation was built using the ISCON simulator (Fancy, 1985). The network has about 100 units. The relevant portions of the network are shown in Figure 12.3 and 12.4. Sequencer nodes are not shown here, since they would clutter the network. The value of the potential decay rate

![Diagram of network](image)

**Figure 12.4.** Section of the network handling noun phrases.
δ used for the simulation discussed below is 0.6. This value was arrived at as an acceptable value after considerable experimentation.

Initially the input units corresponding to the conceptual-level description of the sentence to be generated are activated. For example, in order to generate the sentence *A monkey is eating a banana*, we activate the units corresponding to concept-eat, concept-monkey-1, and concept-banana-1. We also activate the constraint-specification units—current-voice-is-active, and current-aspect-is-continuous. The units global-subject-enable are also activated and the links from concept-monkey-1 to its subject-binder (viz. e-s-1) and from concept-banana-1 to its object-binder (viz. e-o-2) are enabled.

Activation of the concept-eat unit activates the realization-class unit svo. This, along with the fact that the current-voice-is-active unit is on, turns on the choice unit active-svo-choice, which, in turn, sequentially activates the three constituent units connected to it, viz. deep-subject, active-svo-verb, and deep-object.

Meanwhile, activation of the concept-level unit concept-monkey-1 had turned on the realization-class unit individual-item, which feeds an activation link to the binding-completion-signaler node, which however, remains inactive. The binding-completion-signaler unit, which is not shown in figures, is used to indicate the time when binding of subject or object roles to appropriate concept units is complete. Among the constituent units, the deep-subject unit is activated first. Now, the set of units that participates in the binding process comes to play its role. Since deep-subject unit is active, the unit e-s-1 receives activation from all its inputs, and this results in the binding of the concept-monkey-1 unit to the subject role. Finally, as a result of binding, the np-expansion-driver unit provides activation to the binding-completion-signaler unit, which becomes active. It feeds the activation link to the choice unit indef-np, which is turned on. It provides excitatory input to the constituent units det and nounp, which are activated, resulting in the sequential activation of the units word-a and word-monkey. Expansion of the subject is now complete. A special sequencer node is activated; it rests the various nodes involved in the expansion of the subject, making the np subnetwork available for reuse for the object phrase expansion, and also starts the expansion of the verb phrase by sending an excitatory signal to the active-svo-choice constituent unit.

The expansion of the verb phrase of the sentence (or activation of various units in the active-verb-phrase subnetwork, in this particular case) leads to the utterance of the word units word-is and word-eating in sequence. All units involved in the expansion of the verb phrase are turned off, and expansion of the noun phrase corresponding to the object of the sentence is initiated, resulting in activation of the word units word-a and word-banana in sequence. This completes generation of the whole sentence and is followed by deactivation of all units that took part in the process of generation and are still
Figure 12.5. Graphs of potentials of relevant units.
active. Finally, a specialized sequencer unit is turned on. Activation of this unit can be used to generate an appropriate pause (in case of speech) or the punctuation symbol ',' in case of written text. At this point, this specialized node can be used to send an activation signal to all nodes in the network at the reuse-site, which is present in all nodes, to prepare for presentation of the conceptual specification of the next sentence to be generated. By this time, most nodes (except for the ones which were used last in the generation of the current sentence and had not had enough time to decay) would be in inert state, and the reuse signal would force them into the initial state.

A graph showing the potentials of the concept, restriction-specification, and the word units for this example is shown in Figure 12.5. This clearly shows that the words that constitute the sentence are activated in proper sequence. Each word unit is active for a short period of time during which it is assumed to be spoken or written. Examples of other sentences that can be generated by our system include A monkey has been eating a banana, A banana is being eaten by a monkey, A baboon was peeling a banana, etc.

8. AUGMENTING THE MODEL FOR SIMPLE LEXICAL CHOICE

8.1. General Pragmatic Considerations

In this section, we present a simple approach to handling choice of lexical tokens in the generated sentences. Lexical choice is governed by a large number of factors—syntactic, semantic and pragmatic. These include considerations such as relevant characteristics of the speaker and the hearer, their interpersonal goals, the conversational setting, etc. Identification of such factors, and a careful analysis of the nature of their interplay, is essential in order to develop a systematic theory of lexical choice. We hypothesize that different factors provide different levels of evidence for the selection of competing words, and these have to be appropriately combined in order to select a particular word over its competitors.

The existing approaches to lexical choice in natural language generation assume a simple model of the world, based on objects and classes of objects (perhaps arranged in a taxonomy), relations between objects, actions, states, etc. Each of these has associated with it a particular word which is chosen when it is used to produce a sentence. MYCIN (Shortliffe, 1976) and TEXT (McKeown, 1985) use such an approach. More sophisticated systems allow for a choice among terms based on some fixed conditions. For example, the BABEL generator (Goldman, 1975) turns an ingest action into eat if the object is food, but into breathe if the object is air. Similarly, Danlos (1984) chooses assassinate when the victim is famous, murder otherwise. However, no system except Hovy's PAULINE (Hovy, 1986) attempts to consider the effects of pragmatic considerations
in the choice of words and other syntactic decisions. Ward (1988) presents a large number of complex issues—syntactic, pragmatic, and design oriented—which govern the choice of words, and mentions a system under development that will attempt to address many of them, although the details available are very sketchy.

Making a program choose among competing lexical items requires identifying the choice points at which this information can be incorporated, and defining criteria by which to make choices. In this section, we discuss how such a generator may represent relevant pragmatic concerns: characteristics of the hearer, the conversational setting, and interpersonal goals of the speaker and the hearer, etc. We want to emphasize that our intention is not to underplay the effect of nonpragmatic factors on word choice. Our singular aim concerns illustrating through simple examples the suitability of an evidential model for performing word choice decisions. Once we can demonstrate the usefulness of the connectionist model in simple environments, the mechanism can be augmented to incorporate other factors without substantial modification, apart from having to construct a bigger network.

8.2. Particular Pragmatic Considerations for our Model

Though pragmatic aspects of conversation help determine a speaker’s text, their effect is not direct. Attempts to formulate rules that relate pragmatic aspects to generator production decisions often result in rules which are extremely ad hoc. The solution strategy adopted by Hovy (1986) is that he proposes an intermediate set of goals between the pragmatic aspects of conversation and the syntactic decisions a text producer has to make. He calls them rhetorical goals. Various rhetorical goals contain strategies that give rise to stylistic differences in the text, enabling the speaker to communicate various types of additional information to the hearer. In this chapter, we take only one such rhetorical goal into consideration, namely, formality; it has direct effect on the choice of words.

Again, our objective in this chapter is not to elaborate upon the whole gamut of pragmatic issues involved in the complicated phenomenon of lexical choice. We intend to pursue only one rhetorical goal and demonstrate our ability to encode it in an evidential framework. We leave the encoding of other rhetorical goals individually, and the complex interactions among them to further research. However, our belief is that addressing one issue in detail is an appropriate stepping stone towards extensive study and implementation covering other rhetorical goals and accompanying issues of interactions among them.

The various rhetorical goals identified and studied by Hovy (1986) include partiality, detail, formality, haste, etc. Partiality determines the strength of affects introduced into the text by a program. It can be expressed explicitly, or via techniques such as phrasal juxtaposition and stress words. The amount of details
provided may also influence the selection of words during generation. If the speaker wants to be concise, he or she may discard detailed descriptions and may choose phrases which are more succinct and more expressive. Again, the amount of time the speaker allows himself or herself to produce the sentences plays an important role in lexical choice. These factors, and a host of others, affect, not only word choice, but every facet of generation including the sentential and phrasal structure, among others, and have been discussed adequately in (Hovy, 1986). However, we confine ourselves to the manner in which these factors influence word choice only for the purposes of discussions in this chapter.

Formality underscores the dignity and social standing of a word. Colloquial words are appropriate for informal conversation, whereas they are out of place in a dignified and formal utterance. For example, it would be foolish to welcome an elder statesman by complimenting him on being a “wise and venerable guy,” since it entails using a colloquial word in a formal situation. Similarly, many literary and highfalutin terms sound absurd in a colloquial context. We do not praise a friend for dexterity or for his erudition, not, at least, when we meet him on the street or chat with him across the table.

Hovy uses a large number of criteria to decide on the proper level of formality, such as atmosphere, depth of acquaintance, relative social status, desired effect on interpersonal distance, etc. In order not to get cluttered in details, and for the purposes of illustrating the ideas with clarity, we consider only two pragmatic factors—atmosphere and depth of acquaintance. Atmosphere can take two values—formal and informal. Depth of acquaintance can take either of three values—stranger, acquaintance, and friends. There are three rhetorical goals—formal, informal, and colloquial—corresponding to the three categories into which words can be classified based on levels of formality. The corresponding rhetorical goals are called RG:formal, RG:informal, and RG:colloquial respectively.

Below, we attempt to model the effects of formality on word choice in a connectionist network, in terms of units and weighted connections among them. We introduce two levels of units—the rhetorical goal level, and the criteria level. There is a node corresponding to each value of rhetorical goal and each value of the two pragmatic factors under consideration. In order to do this in a systematic manner, we have come up with numerical values representing the strengths of association among the various rhetorical goals and the criteria that govern the choice of these rhetorical goals. We can have all possible weight levels between 0 and 1, but in such a situation it will be difficult to justify the choice of weights, e.g., why did we not choose 0.4 instead of 0.45, etc. It is not possible to do an empirical statistical study to come up with these precise weights. Hence, we have decided to keep the number of different activation/inhibition weights on links as low as possible. Currently, the different levels we have are: high activation, low activation, high inhibition, low inhibition, and unbiased support.
The strengths of connection among the various units discussed thus far is given in the table in Figure 12.6. If we examine the first row, the formal-atmosphere node has a high activation link to the rhetorical goal unit RG:formal. In other words, if we have a formal occasion, our tendency to use formal style in the choice of words is high. Similarly, the second entry in the first row specifies that, in a formal occasion, one’s tendency to use informal choice of words is moderately low. Finally, the third entry in the first row reflects our belief that a formal atmosphere gives strong support against one’s choice of colloquial word style. Thus, in our evidential framework, nodes corresponding to different values of various affecting parameters provide different levels of evidence towards the choice of word style. The units used for deciding on a level of formality and their interconnections are shown in Figure 12.7. The derivation of the weights on the various links in this figure is discussed later in this section. An example network to illustrate how this technique is used to choose between competing words is shown in Figure 12.8.

The values of high and low activation/inhibition shown in Figure 12.6 have been chosen by experimentation to be:

Low inhibition = -0.4  Low activation = 0.4
High inhibition = -0.8  High activation = 0.8

8.3. Implementation and Simulation

In a single sentence, one usually uses words on the same formality scale. However, from sentence to sentence, one may deliberately choose different formality
scales; this may happen, not because of change in one’s situation, but because of a change in one’s feelings. One may speak formally to one’s family or most intimate friends when one is angry or embarrassed (Shopen & Williams, 1980). Somebody explaining a highly technical process may use very informal terms like *this little doodad* to lighten the terminological load a little. Teachers or preachers may try to show young people that they understand their problems and can still “speak their language.” But, in most cases, mixing of formality levels of styles leads to funny, unacceptable sentences; the clash stands out very clearly. Silva and Zwicky (1975) have suggested that degrees of discord could be measured by assigning a number to each element of a sentence—formal elements
between 0 and 10, informal ones between 0 and -10, etc. The degree of stylistic
deviance of a sentence is measured by the difference of values in its most extreme
elements—a sentence that contained both a -10 item and a +10 item would
have the highest possible discordancy value, 20. Such a sentence would probably
sound ludicrous, while a sentence with a discordancy value of 2 to 5 might sound
relatively normal. Although the approach sounds simple, assigning values is
quite difficult, and the values assigned may change from person to person. Based
on this and other observations by linguists, we have decided to use one level of
formality in all the words in a sentence. If we want to use different levels of
formality appropriately in different words, it will be a very difficult task to
determine the factors that influence such decisions, the expressions/words that
need to be used, etc. Therefore, we choose a formality level at the beginning of
the sentence production process (of course, in parallel with other activities re-
quired for production) and thereafter derive the choice of words based on the
predetermined formality level.

We do not give equal importance to the two categories of criteria. We hypo-
thesize that the atmosphere has stronger influence on the choice of word formality
level than the level of acquaintance. In other words, atmosphere is a more
prominent determiner of word formality level; atmosphere units provide a stron-
ger activation/inhibition to the rhetorical goal units. In order to reflect this, we
multiply atmosphere strengths by a factor of $\delta_1$, and those of level of acquain-
tance by another factor $\delta_2$ such that $\delta_1 > \delta_2$. For our implementation, the values
used were found by experimentation to be $\delta_1 = 1$ and $\delta_2 = 0.6$

It is sometimes not possible to decide, based on the value of only one of the
affecting criteria, what level of word style is to be used. For example, in the
second row of the table in Figure 12.6, there is no clear-cut choice for/against
using informal or colloquial word style when the atmosphere in which generation
is done is informal. In such a situation, we use a weight level, which we call
unbiased support. It is true that informal atmosphere provides evidence towards
choice of either informal or colloquial style. Therefore, we must have activation
signals going from informal-atmosphere unit to all RG:informal and RG:collo-
quial units. However, since the weights of these connections cannot be deter-
mined properly, we provide equally weighted connections to each one of these
nodes. The strength of such a connection is $1/N$ where $N$ is the number of units
receiving unbiased support. Since in our example, there are two such units, the
connection weight on each is 0.5.

Figure 12.9 shows the potential of the rhetorical goal units as they compete
among themselves when the units depth-of-acquaintance = friends and atmo-
sphere = informal are activated. Only two rhetorical goal units—RG:colloquial
and RG:informal—become active; since RG:formal does not receive any posi-
tive activation, it does not become active. At first, potential of both units rises.
After a while, the potential of RG:informal starts falling, whereas the potential of
RG:colloquial continues to remain high.
We note at this point that the following three sentences have the same conceptual content; only the level of formality is different.

- The gentleman addressed the lady.
- The man talked to the woman.
- The guy chatted with the gal.

As shown in Figure 12.9, when depth-of-acquaintance = friends and atmosphere = informal, RG:colloquial wins over RG:informal. Figure 12.10 shows the potential of various word units—the manner in which competition among contending lexical items takes place. We consider verb—preposition pairs such as talk—to, chat—with, as one word unit each. The clock cycles in Figures 12.9 and 12.10 are same. We mention that the rhetorical units take up to 10 units of time to settle, and the production of the first word in a sentence does not start till later than that. Therefore, it is not necessary to delay the production of the phrases in the sentence till after the stabilization of rhetorical goal units.

9. DISCUSSION

In this chapter, we have demonstrated that the connectionist model of computation can be used for the generation of English sentences from an input specifica-
tion. We achieve this by using several levels of interacting units, where each level corresponding to a well-defined step in the process. Although simple, we have shown that such a stratified model can be successfully used for the production of simple sentences in English. Also, it has potentials for extensions, which we discuss in brief in the following paragraphs. We also present some of the shortcomings of our system below.

The sentences generated by the current network are all of the form where a sentence contains a subject, a verb and an object. They can be in active or passive voice and in several different tenses. All noun phrases generated have a determiner followed by a noun. All verbs used so far are transitive. Also, all passive sentences currently generated have the by-phrase following the verb. We need to incorporate intransitive verbs and allow the agent phrase to be optional in passive sentences. Further research can be directed at many other fronts, such as incorporation of relative clauses, repeated constituents such as prepositional phrases and adjectives, anaphoric pronominal phrases, etc. Gasser (1988) has addressed some of these issues within his framework, although he fails to present a principled discussion on why such constituents are chosen and how they are actually generated.

An open problem that needs to be addressed is the matter of mechanisms for interfacing our surface generation system with other systems that perform content determination and text planning. Our approach to specifying this interface is simple but inefficient. A more sophisticated technique might involve using exploded case frames, introduced by Cottrell (1985). This will allow us to prevent the generation of such absurd sentences such as A banana is eating a monkey by imposing selectional constraints on the fillers of roles of the verbs. This will
necessitate the incorporation of an elaborate knowledge base describing the concepts that are available in our domain of interest. The specification and organization of such a knowledge base such that responses to pertinent queries can be elicited efficiently can be modeled after Shastri (1985). Additionally, in the current implementation, we need two binder units—a subject-binder and an object-binder—for each concept that can fill the subject/object role of a sentence. Since the number of such concepts is usually very large, it will be worthwhile to investigate other approaches to the problem of binding. In general, this problem is an instance of the general problem of binding in connectionist systems. This problem is not specific to our work, and any elegant solution to it will be applicable here.

At this point, it will be pertinent to point out that Gasser’s (1988, p. 82) basic technique of binding simple concepts to conceptual roles of an action concept consists in creating intermediate units (object, subject, etc.) and connecting the action instance concept (e.g., transfer-5) to the object/subject concept through this intermediate node. This approach is not dissimilar from ours, where specific links are activated to identify concepts with various roles in order to achieve binding, before the process of generation starts. Gasser’s (1988, p. 99) solutions to cross-talk are not very different from ours either. In our case, if at the choice level a decision has been made to use the svo form, we know the sequence of components is subject phrase, verb phrase, and object phrase, and can use sequencing circuits to activate expansions of different phrases sequentially. If the form vso (a question form) was chosen, the sequence would be the verb phrase followed by the subject and object phrases, respectively. This sequencing is forced automatically by the circuit once the appropriate choice unit has been chosen. Gasser attempts to achieve the same goal using either differently weighted links to constituents (he does not specify how these differential weights can be determined), or by using a specialized type of WTA network (applicable only for a small number of roles).

Finally, we have discussed how the formality level affects the choice of words during the process of sentence generation. However, we have considered only two of the criteria that affect formality. It will be worthwhile to study how the other factors can be incorporated in an evidential network. Additionally, we need to study how characteristics other than formality affect word choice and investigate them with regard to implementability.

REFERENCES


