Lexical Access and Serial Order in Sentence Production

Joseph Paul Stemberger
Carnegie-Mellon University

Serial ordering in the perception and production of language is a phenomenon that is no longer taken for granted in cognitive science. In the serial models that were in vogue until recently, it is a simple enough phenomenon, and a relatively easy thing to encode. In the parallel interactive models that are currently receiving much attention, however, it turns out to be a very difficult thing to encode (McClelland and Rumelhart 1981, Rumelhart and Norman 1982, Dell and Reich 1980). Problems arise in most models despite the relative simplicity of the tasks that they model; in perception, serial ordering is available in the input, while in the production of words in typing and speech, the serial order is available as a schema made available by each accessed word. The ordering of words in sentences is a far more difficult serial-ordering task, because sentences are not stored as units; the speaker cannot take a semantic/pragmatic representation of the sentence and use it to access a schema for that particular sentence. Rather, s/he must use more general strategies for ordering the words. In this paper, I will present one possible solution to the problem of how words are ordered in the production of sentences. There are several distinct problems that must be solved: accessing exactly the right number of words for the sentence, getting those words in the right order, and what to do when a word appears more than once in the sentence. These problems will be discussed in turn.

First, exactly the right number of words must be accessed. This is not guaranteed by the semantic representation. Consider a sentence such as _It has a very nice taste/flavor_. In this case, there are two words which are virtually synonymous in the context. On the basis of meaning, it is likely that both words could be accessed, and the speaker would say both, one after the other. How can this be prevented? We must hypothesize that the speaker creates a specific number of positions, or SLOTS, for a sentence. We can view words as being associated with these slots, with the prohibition that only one word will normally be associated with a single slot. All the words that are activated by the semantics/pragmatics will be in competition with each other and inhibit each other. The slots provide a source of activation to a word that enables it to overcome the inhibition from other words. There will thus be as many words accessed as there are slots providing this source of activation.

These slots can be used to determine the serial ordering of the accessed words. We can assume that the slots are controlled in a hierarchical fashion that corresponds to surface syntactic phrase structures. For example, there would be units such as S that control the access and ordering of units such as NP and VP. Slots such as N and V control the access and ordering of words. All serial ordering can be derived from such hierarchical structure, in a way similar to the typing model of Rumelhart and Norman (1982). We must assume that there is a threshold for the execution of a word, and that words reach that threshold one by one; execution in this order consitutes serial ordering. In the Rumelhart-Norman model, words provide a template that determines the activation levels of the letters to be typed. Specifically, the letter at the beginning of the word keeps all later letters inhibited, the second letter keeps all later letters inhibited, and so on. The result is a pattern of activation where the initial letter is highest in activation, and the activation levels decline the later in the word the letter is located. In production, the activation levels of all the letters gradually rise, and each is typed as it reaches a threshold. Thus, the letters are typed in the proper order. This scheme will be adapted for syntax with two changes: control is hierarchical, and the system uses only activation for serial ordering, not inhibition. Some units are differentially activated, and hence reach the execution threshold earlier than others. In execution, a large burst of activation is given to the highest node, S. That node passes activation on to its daughter nodes, but passes the activation at a faster rate to one daughter than the other; usually NP gets the activation at the highest rate in English. NP passes the jolt of activation on to its daughters, with DET getting activation at a faster rate than ADJ, and ADJ at a faster rate than N. These nodes will also pass the activation on to the associated lexical items. Thus, the lexical item that reaches execution threshold first will be the determiner, then the adjective, then the noun, then the verb, and so on. In this way, the lexical items of the sentence will reach the threshold for activation one at a time, in the right order. Feedback from nodes and words to their associated higher nodes is obviously important. This feedback will serve to increase the activation levels of the associated higher nodes, and that extra activation will be passed on to the later daughter nodes. Since a system of this sort must be set up so that the amount of activation passed by a single connection decreases with an increasing number of connections coming into a node (or, equivalently, that the threshold of such a node is increased), it will be the case that a node with many daughters will get less feedback in a unit time than a node with few daughters. As a result, its activation level will rise more slowly and pass on less activation to both higher nodes and later lower nodes. The activation of later nodes, then, and their associated lexical items, is slowed. Thus, through feedback, the system is sensitive to the length of embedded material; later material will be slowed up if there is a lot of material in the subject NP, for example, and will not tend to be executed early.

This leads us to the second problem. Given that a specific number of slots are activated and that all the right words are accessed, how can we ensure that the words become associated with the right slots, so that they appear in the right order? It is necessary and useful to assume that syntactic slots have conditions on them. Each condition can be viewed as a feature that will give a certain amount of activation to specific types of words. This extra source of activation for words will make it more likely that a particular kind of word will be accessed. For example, an N node gives activation to nouns, making it very likely that a noun will gain enough activation to be accessed, and making it harder for verbs to gain enough activation to be accessed. Other features on nodes probably include tense, aspect, person/number/gender marking, and case. This addition to the model ensures even more accurate access of words, but does not ensure complete accuracy. For example, if two words share many of the same features, such as two nouns do, it does not guarantee that the right word will be associated with the right slot. Further, it does not guarantee that a noun (e.g. destruction) will not be accessed by mistake for one of the noun positions when the target verb is closely related semantically (e.g. destroy).

To ensure absolutely that a word is associated with the proper syntactic slot, we must posit the existence of a very special type of feature. This feature will code words and slots for some type of role (Bock 1982). This role may be meaningful, such as the semantic or syntactic role of the information in question. It is possible, however, that it may just be an arbitrary feature that is assigned to a given chunk of the semantic/pragmatic structure on a nonce basis. In any event, role codes which units go together, including which slot goes with which word. Every unit, whether a word or a slot, must have several copies, differentiated by role. Role will function like any other feature; if the noun slot is role-1, for example, it will give more activation to role-1 nouns than to any other nouns, generally ensuring that the role-1 noun will be accessed in that position. The extra activation to lexical items that comes from this role feature will thus ensure that the words that are chosen will not only be the right number and of the right syntactic categories and inflected in the proper way, but will also be of the right role. Figure 1 illustrates how this scheme might work in the context of a simple sentence like the dog ate the apple; arrows represent connections that pass activation, while filled circles represent inhibitory connections. An associated chunk of material on the semantic level will give some activation to a lot of lexical units. However, the most highly activated units will be those that are connected with the most activated semantic units, here the copies of the words dog and apple; inhibition between pairs of lexical units will remove all other nouns from the competition. Activation from the role units on the semantic level will give the role-1 copy of dog and the role-2 copy of apple the highest levels of activation, and all other copies of these words will be inhibited. In parallel with this activity, syntactic nodes have also been accessed. The role-1 N node will become linked to a role-1 noun, here dog, while the role-2 N node will become linked to a role-2 noun, here apple. These N nodes will be linked to higher syntactic nodes in such a way that the role-1 node and its associated lexical unit will precede the role-2 node and its associated lexical unit. Thus, this system will access the right lexical units, associate them with the right syntactic positions, and assign them the correct serial order.

The last problem to be solved is what happens when a word appears more than once in the sentence. It cannot be the case that a word is simply linked to two slots. In the serial ordering scheme that we are assuming, such words would wind up being executed only once, and far in advance of where they should be, since the word will simply sum the activation coming from the two slots and reach execution threshold early. It is necessary to assume that there are several copies of the word available, and that different copies will get accessed for the different positions in the sentence. Since we assume that there are several copies available with different roles, this presents no problem. The solution to the previous problem solves this problem as well.

The system as described here does not represent simply an arbitrary solution to the problem of lexical access and serial ordering. Rather, it has been taylored to account for data from errors which occur spontaneously in natural speech. Stemberger (1982) reviews many of the important properties of speech errors. First, errors where the wrong word is accessed but is related to the target word semantically or phonologically generally are constrained by the features on the syntactic slot; i.e., they are of the same syntactic category and are inflected in the same way. Second, words can be accessed correctly but in the wrong syntactic position(s). Such words are generally of the same syntactic category as the words that they replace, but are often not; the semantic activation of such words means that they can occasionally be accessed in a given position without the help of all the features on the slot. Such words tend to be inflected like the target word, however, even if they are of a different syntactic class from the target word. Such accessing errors lead to anticipations, perseverations, and exchanges. Exchanges are viewed as complex errors, where two words are accessed in the wrong slots. It is possible that exchanges occur more than might be expected by chance. If this is the case, then we must assume that the copies of a single word with different roles have greater levels of inhibition between them than nonrelated pairs of words have, making it harder to use a given word twice in the same sentence. If a correct word is accessed in the wrong slot, this will make exchange errors more likely than chance, since it will be harder to access the word in the correct slot as well. Such inhibition will make it most likely that the other correct word will be accessed in the other slot. However, occasionally some other word entirely may be accessed in that slot, generally a word that is related to the target word; such errors are called "bumper cars" by Stemberger (1982). Various other types of errors are also predicted that do occur, such as partial assimilations and sequential blends. As the model predicts, there are also frequent errors, known as "shifts", where two adjacent words flip around and are executed in the wrong order, as one word reaches threshold prematurely. This description of lexical access accounts for a wider variety of the types of speech errors that occur during lexical access and serial ordering than any other model proposed to date.

The problem of accessing words and putting them in the right order is not trivial. I have presented a preliminary solution to the problem, and one that accounts for much of the known data on problems that arise during lexical access. Much work remains to be done on working out the full ramifications of this model, and on implementing it.

REFERENCES

Bock, K. Toward a cognitive psychology of syntax: Information processing contributions to sentence formation. Psychological Review, 1982, 89, 1-47.

Dell, G., and Reich, P. Toward a unified model of slips of the tongue. In V. Fromkin (ed.), Errors in linguistic performance, pp. 273-286. New York: Academic Press, 1980.

McClelland, J., and Rumelhart, D. An interactive activation model of perception, Part 1. Psychological Review, 1981, 88, 375-407.

Rumelhart, D., and Norman, D. Simulating a skilled typist: A study of skilled cognitive-motor performance. Cognitive Science, 1982, 6, 1-36.

Stemberger, J. The lexicon in a model of language production. Unpublished doctoral dissertation, UCSD, 1982.

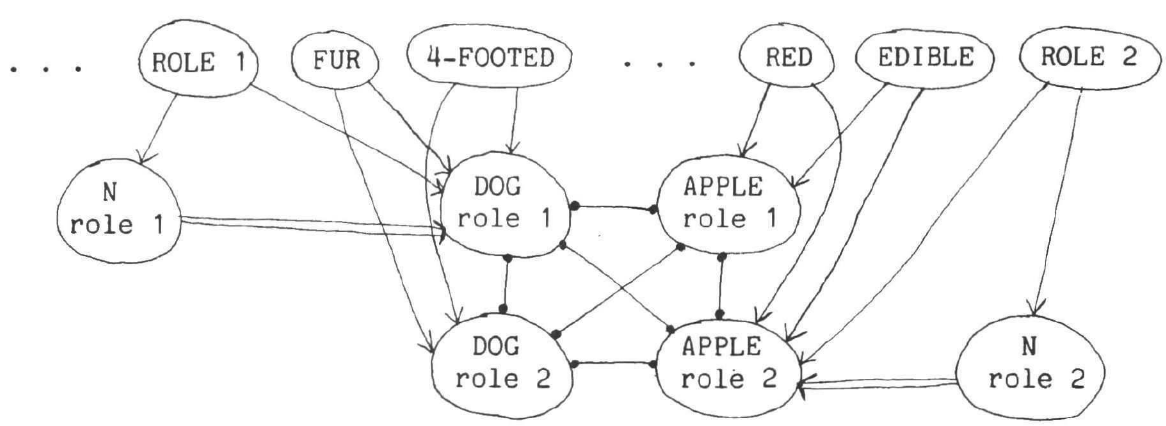

FIGURE 1: Access of nouns in the sentence "the dog ate the apple".