Issues in Word Choice

Nigel WARD

Computer Science Division
University of California
Berkeley, California, 94720, USA

Abstract

This paper discusses word choice for natural language generation. It examines 11 issues, the solutions that have been proposed for them, and their implications for design. The issues are:

- How are appropriate words chosen?
- How is conciseness ensured?
- When does choice stop?
- How are patterns of lexicalization respected?
- How are interactions among choices handled?
- How are the correct parts of speech chosen?
- How are words chosen to satisfy constituency?
- What ensures that a word stands in the correct relation to its neighbors?
- How is word order determined?
- Are all words chosen in the same way?
- In what order are the factors considered?

This paper also discusses FIG, a generator which incorporates novel solutions to many of these issues. FIG violates common assumptions about the roles of modularity and grammar in generator design. Analysis of FIG leads to 4 principles for generator design, as follows:

- Have an explicit representation of the status of the generation process at each point in time.
- Use a single, unified representation.
- Do not rely on the details of the structure of the input.
- Treat most choices as emergent.

1. Word Choice and Generator Design

An important task in the generation of natural language is choosing words. This paper presents issues in word choice. A generator must handle these issues or risk producing output which is inappropriate, unnatural, confusing, unreadable, or ungrammatical.

Choice has been called the key problem in natural language generation /McDonald 1983/. However, most research so far has focused on syntactic choice; word choice has received little attention /Pustejovsky and Nirenburg 1987/. This paper focuses on basic issues in word choice and their implications for generator design.

2. Overview of FIG

Before discussing the issues, I briefly present "FIG," my generator. This is necessary because FIG handles many issues in ways which are not discussed elsewhere in the literature.

FIG, short for "Flexible Incremental Generator," was designed to be useful for both machine translation and cognitive modeling. It is based on the idea that speaking is a process of choosing words one after another. It has been incorporated into a prototype Japanese-to-English machine translation system. An example of its output is:

(1) *"One day the old man went to the hills to gather wood and the old woman went to the stream to wash clothes."*

Thanks to Terry Regier, Dan Jurafsky, and Robert Wilensky. This work was supported in part by a Sloan Foundation grant to the Berkeley Cognitive Science Program and by the Defense Advanced Research Projects Agency (DoD), Arpa Order No. 4871, monitored by Space and Naval Warfare Systems Command under Contract N00039-84-C-0089.

Processing Characteristics

1 Each node of the input conceptualization is a source of energy.

2 Energy flows through the semantic network.

3 The currently most highly activated word is chosen and emitted.

4 Activation levels are updated.

- This four-part cycle repeats until all the input has been conveyed.

An utterance is simply the result of the successive choices of words. Thus FIG is an incremental generator.

Representation Characteristics:

A single semantic network represents world knowledge and language knowledge. FIG uses a variant of Cognitive Representation Theory /Wilensky 1987/. The key characteristic for generation is that this representation is a semantic network which includes language knowledge, after Jacobs /Jacobs 1985b/. In particular, the network includes nodes for concepts, words, syntactic features, constructions, and constituents of constructions. (Node names are henceforth set in bold and preceded by a single quote.) The links among nodes represent associations in world knowledge and language knowledge. In particular, there are links from concepts to words that express them.

The energy level of a node represents its relevance at each point in time. A "relevant" word is one which could form part of the output, a "relevant" construction is one which could provide an appropriate structure to the output, and a "relevant" concept is one which is associated with the meaning to express. So that activation levels represent the *current* relevance of nodes there is an update mechanism. After a word is output, this mechanism: zeroes the energy of the word just emitted, zeroes the energy of that portion of the input which has been conveyed, and for each construction, zeroes the energy of constituents which have been completed.

Energy flow across links represents evidence for the relevance of a node. The energy level at each node is given by the sum of the energies reaching it from other nodes.

To see how FIG chooses a word, suppose the input includes nodes like **'woman**, **'old**, **'live**, and **'day**; and that syntactic considerations are currently activating verbs. Then **'"live"** will have the highest activation and be emitted next. It will have more energy than any other verb, since it also receives energy from the input; and it will have more energy than any other word suggested by the input, since it also receives energy from **'verb**. Thus, FIG will emit *"live"* next. One can say that FIG is equally syntax-directed and semantics-directed.

This brief discussion omits aspects of FIG of no direct relevance to word choice. Much more could be said about the exact activation algorithm, the representation of constructions, the role of link weights, the use of instantiation for utterances involving more than one occurrence of a word, and so on.

3. Basic Issues

Each of the following points is illustrated by examples of output which a generator should not produce.

Issue 1: How are appropriate words chosen?

A generator must choose words appropriate for the input that it is called to express. There is not much interesting to say about the simple case, in which a word is "appropriate" if it refers to some "concept" of the input meaning. This simple case can be handled with a dictionary mechanism to look up the word for a concept, or with a mechanism to traverse the link from a node to the word.

However "appropriateness" is not always so simple. Three complications are discussed at length below, but first, it should be noted that many researchers avoid these complications. They do this by considering them to be problems of "concept choice," not word choice. This leads Thompson, for example, to postulate a pre-processor, a "strategic component," whose output only includes concepts which map easily to words /Thompson 1977/.

Complication 1: The relation between a word and the input can be complex. For example, diverse facts about the input rule out:

(2) *"drink soup"*

if the soup is eaten with a spoon rather than by sipping from the bowl (versus *"eat soup"*)

(3) *"she went to the river"*

if it was narrow, fast-moving, low-volume, etc (versus *"she went to the stream"*)

(4) *"he went to the hills"*

if the distance traveled was short, and he planned to stay in the hills for a while and move around there (versus *"he went into the hills"*)

(5) *"he met her in the bus"*

if the bus was running in scheduled service /Fillmore 1985/ (versus *"he met her on the bus"*)

How can a generator choose words which depend on more than one element of the input? There are several answers:

Goldman /Goldman 1974/ analyzes words with complex meanings as having a core meaning plus conditions on use. For example he considers INGEST to be the core meaning of the word *"drink."* This reduces the problem of word choice to the problem of choosing among the various words associated with a core element. Goldman's BABEL chooses by testing nearby nodes. For example, for INGEST it tests whether the object of ingestion is liquid in order to decide whether to use *"drink."* As Danlos points out, the organization of tests into discrimination networks "is bound to be arbitrary" /Danlos 1987/.

After finding candidate words (explained below) Hovy's PAULINE /Hovy 1987/ matches the meaning of the word to the input to determine if the word is appropriate.

In FIG words with complex meanings are simply suggested by more than one factor. For example, if the input includes nodes like **'liquid** and **'ingest**, then **'"drink"** receives activation from both of them. This gives it high cumulative activation, which makes it likely to be chosen.

Complication 2: The relation between a word and the input can be tenuous. For example, the words of a paraphrase can be "appropriate" even if they do not directly correspond to any element of the input. Hovy gives the example

(6) *"In the primary on 20 February, Carter got 20515 votes. Kennedy got 21850."*.

and comments, "if we want good text from our generators, we have to give them the ability to recognize that *"beat"* or *"lose"* or *"narrow lead"* can be used instead of just the straightforward sentences." How can a generator choose such words?

Hovy's PAULINE finds a set of "candidate" topics by considering concepts related to the input nodes and also concepts which serve rhetorical goals. These topics then map to words.

Jacobs' KING /Jacobs 1985b/ "searches" through world knowledge to find words. The search process only crosses links of certain types, which ensures that it only reaches words with equivalent meaning, such as *"buy"* for commercial-transaction.

In FIG a word can receive energy even if it is not directly linked to a node in the input, via the links of world knowledge. This is simply a case of priming by memory associations. Activation attenuates every time it crosses a link, which ensures a bias in favor of words which are "nearer" to the nodes of the input.

Complication 3: The input to a generator can include more than meaning. Consider:

(7) *The stream was the place where the old woman went.*

This utterance is strange, unless the stream is to be highlighted. In general, word choice can depend on the relative importance of the portions of the input and on the way the input is "framed" /Fillmore 1985/. How can these factors affect word choice?

No existing generator seems to consider these factors when choosing words. However, certain architectures seem more open to such factors. Generators with "open" architectures include Jacobs' PHRED /Jacobs 1985a/, which allows hashing on any factor for lexical access; and FIG, in which any factor can be a source of activation.

Issue 2: How is conciseness ensured?

A generator should not produce

(8) *"a peach located at the surface of the water and supported by the water."*

(versus *"a floating peach"*)

KING's knowledge consists of a taxonomy of concepts /Jacobs 1985b/, so it can simply choose the most "specific" word. Hovy's PAULINE chooses the word whose meaning configuration is "largest," that is, the one whose meaning subsumes as much of the input as possible.

FIG handles this rule without additional mechanism: words with "large" meanings become highly activated simply because they get energy from many nodes of the input. Thus, FIG has an intrinsic bias to use the most specific word possible. For example, if nodes like **'verb**, **'motion**, **'transitive-action**, and **'initially-scattered** are activated, then energy spreads to **'"get"** and to **'"gather"**. However, *"gather"* gets rated as more appropriate, since it receives energy from one more source than **'"get"** does, namely from **'initially-scattered**.

Issue 3: When does choice stop?

This question can be stated more specifically as "when does the generator stop saying things about some topic?" The basic problem is avoiding redundancy.

(9) *She saw a peach floating in the stream, **being moved by the current, and moving downstream.***

This utterance is redundant in that the information given by the words in bold is inferrable from the first clause. It should be noted that many researchers avoid this issue by assigning it to a pre-processor. This allows a generator to simply express all the nodes or propositions present in its input — implicitly preserving the amount of information.

FIG models inferrability with a simplified version of Norvig's marker-passing scheme /Norvig 1987/. Each time it chooses a word it "marks" the parts of the input which the reader can now infer. For example, after the words *"gather"* and *"wood"* are emitted it marks the **'gather-firewood** node, representing the fact that that script has been conveyed. Only the unmarked input, representing the information that still needs to be said, is a source of activation. FIG terminates when it has marked all of the input.

Issue 4: How are patterns of lexicalization respected?

A generator must prefer words which belong to the lexicalization patterns of the target language and genre. This issue has not yet been discussed in the generation literature, so I illustrate it with examples of output which violate lexicalization patterns.

(10) *"he **entered** the cellar **running**"*

(versus *"he **ran into** the cellar")* There is a general preference to conflate motion and manner into the verb /Talmy 1975/.

(11) *"his **reliance** on it was excessive"*

(versus *"he **relied** on it too much"*) Actions are better expressed as verbs than as nominalizations, other things being equal. In general, there is a preference to use words which are of the correct part of speech for a given semantic need.

(12) *"he **has stood up**"*

(versus *"he **is** standing"*) States are best expressed by describing them, rather than by using the cause or the onset metonymically /Talmy 1985/.

(13) *"let's **eat** at a restaurant"*

if the context is "what shall we do now?: (versus *"let's **go** to a restaurant"*) Complex actions are best expressed by mentioning the onset.

(14) *"an old **person** went to the stream and found a **fruit**"*

(versus *"an old **woman** went to the stream and found a **peach**"*) There is a preference to use basic level words and sex-specific words.

No existing generator handles patterns of lexicalization. One possible approach would be to use special procedures: to "carve up reality," for example to specify which information to conflate into a word; and to specify which aspects of a situation to encode, for example, which word to use for a metonymy. Within the FIG framework there are other possible solutions. There could be special nodes like 'words-conflating-motion-and-manner' to give energy to appropriate words, or the relative densities of knowledge about certain concepts could felicitously cause choice of basic-level words.

Issue 5: How are interactions among choices handled?

A generator must not, for example, violate collocations:

(15) *"**high** air currents"*

(versus *"**strong** air currents,"* yet ***high** winds"*). The problem here is that the choice of an adjective can depend on the noun chosen.

The standard way to handle such things is to order choices. For example, heads are chosen first so they can constrain the choice of modifier. Usually the order of choices is fixed by the basic algorithm of the generator. For example, syntax-driven generators choose words in the order that they expand and traverse the syntax tree, and data-driven generators choose words in the order that they traverse the input /McDonald 1983/.

In FIG there is no need to order choices. This is because the mere possibility of using a word can affect other choices. For example, if **'"winds"** seems relevant it will have energy, and this energy will spread to **'"high"**. (Recall that the network has links between associated words.) Other things being equal, such energy will make **'"high"** be more activated than words such as **'"strong"** or **'"fast"**. Thus FIG will produce *"high winds"* but *"strong air currents."*

4. Syntactic Issues

It makes no sense to choose words without regard to syntax. This section discusses some interactions of syntax and word choice. But first, I briefly sketch the syntactic theory which underlies FIG's treatment of grammar.

Construction Grammar is a theory of syntax currently being developed at Berkeley. Construction Grammar "aims at describing the grammar of a language directly, in terms of a collection of grammatical constructions" /Fillmore 1987/. Each construction represents a pairing of a syntactic pattern with a meaning structure. Construction Grammar differs from most theories of language in accounting for the structure of complex grammatical patterns, such as lexically-headed constructions /Fillmore, Kay and O'Connor forthcoming/, rather than focusing on core syntax. It also differs in stressing the dependence of language on other aspects of cognition /Lakoff 1987/.

A construction has "external syntax," which describes where and when it is appropriate; and "internal syntax," which describes its constituency structure. Consider, for example, the Existential There Construction /Lakoff 1987/, as in *"once upon a time there lived an old man"*. Two facts about the external syntax of this construction are that it is used to introduce people or things into a scene, and that it overrides the normal subject-predicate ordering. The internal syntax of the Existential There Construction includes three constituents, roughly the word *"there,"* a verb, and a noun, in that order.

Since Construction Grammar is based on declarative constructions rather than procedural rules, it is well suited to implementation with a network. In FIG constructions and their constituents are nodes of the network.

Syntactic Issue 1: How are the correct parts of speech chosen?

For example, a generator must avoid output like

(16) *"When she got to the stream, **her** saw a peach which was **float** there"*

(versus *"**she** saw a peach which was **floating** there"*)

Syntax-driven generators typically handle this issue by setting up constraints and then finding a word that satisfies them. To use an old term, these generators do "lexical insertion." Syntactic constraints can be manipulated in several ways. For example, a top-down generator accumulates constraints as it works down the tree, and these govern word choice at the leaves of the tree.

In FIG constructions are linked to syntactic features which describe the syntactic characteristics of constituents. This allows activation to flow from constructions to features, and thence to words linked to those features. For example, suppose that **'ex-there**, the node for the Existential There Construction, is activated. Energy will spread from **'ex-there** to the feature **'verb**, and from there to all verbs.

Syntactic Issue 2: How is word order determined?

Word order is not usually treated as a separate issue. This is because most generators handle it implicitly, as they follow through on syntactic choices. They do this by variously expanding trees, traversing networks, or matching templates.

Appelt took a different approach: his planning-based generator manipulated word order explicitly /Appelt 1985/.

In FIG word order is determined by the activation levels of various constituents of constructions. The update mechanism ensures that the activation level of each constituent correctly reflects the current syntactic state. Suppose, for example, that FIG has already emitted *"Once upon a time, there"*. Next it should emit a verb, according to the Existential There Construction. This is represented by having the second constituent of **'ex-there** be highly activated at this time. Energy flows from the second constituent to the feature **'verb**, and from there to all verbs. Thus the activation levels of constituents help determine what word gets chosen and emitted next. This suffices to produce correct word order. In effect, constructions shunt energy to words which should appear early in the output.

Syntactic Issue 3: How are words chosen to satisfy constituency?

Constructions have constituency and words have valence, which a generator must respect.

(17) *"The woman went to the stream. **When got to**, she saw, to her surprise, an enormous peach."*

is bad because verbs require subjects and because *"got to"* requires a destination. This issue is complicated by the existence of optional constituents. For example, consider the noun-phrase construction. Information relevant to an object can often be expressed with an adjective, so that option must be available. But if there is no appropriate information the adjective option must be passed up. The general problem of constituency can be stated as: in what way does syntax affect the decision to use a word or not?

Syntax-driven generators such as PENMAN /Mann 1983/ handle constituency in their basic algorithm. The syntactic structure is determined before word choice is done. A common way to handle optional constituents is by augmenting the grammar with specifications of how to test the input. The results of these tests determine whether or not to use an optional constituent. This of course requires a special mechanism to execute these tests.

FIG simply does not choose words for optional constituents unless they are appropriate. In FIG each word "competes" with every other word in the lexicon to be the most highly activated. In particular, each word is in competition with words which could come later in the utterance. This suffices. For example, suppose FIG has just emitted *"the"*, and, accordingly, **'noun-phrase's** second constituent is highly activated and its third constituent is somewhat activated. From these constituents the feature **'adjective** gets a lot of activation and the feature **'noun** gets somewhat less activation. There are two cases:

1. If the input includes some information expressible with an adjective, then both adjective(s) and nouns will get energy from the input, but an adjective will probably be emitted, since **'adjective** is activated more highly than **'noun**.

2. If there are no concepts which could be expressed with an adjective, then some noun will get energy both from the input and from **'noun**, but any adjective will only have energy from one source, **'adjective**. Thus a noun will probably be emitted next.

Syntactic Issue 4: What ensures that a word stands in the correct relation to its neighbors?

A generator must not scramble words, as in

(18) ***"the green man went to the old hills"***

where the adjectives are attached to the wrong nouns (versus ***"the old man went to the green hills"***).

The most common solution is to use syntax-directed techniques, similar to those discussed under Syntactic Issue 3. The grammar typically specifies the location of information for dependent words. For example, the generator might always follow "modified-by" links to reach adjectives for a noun. A different formalism with the same effect is unification /Appelt 1983/.

Since the input to FIG is a structure of linked nodes related concepts tend to be activated together. This means that FIG has, in effect, a "focus of attention" /Chafe 1980/. For example, if **'old-man37** is activated, then energy flows to related nodes, such as those encoding his appearance, location, and goals. Therefore at the time when **'man** is highly activated (and probably *"man"* is about to be output) nearby nodes, like **'old**, become highly activated.

5. Design Issues

Thus, there are many issues in word choice. Their importance can be questioned — after all, every existing generator ignores many of them, and yet generators have produced outputs which look quite good. However, close analysis shows that this is only because the inputs have been tailored to determine a good sentence. In other words, most generators' inputs are English sentences in disguise. Such generators only have to do the amount of computation needed to retrieve the target sentence. For example, (to oversimplify) Goldman's BABEL /Goldman 1974/ really only had to choose among words with some common element of meaning; McDonald's MUMBLE /McDonald 1983/ really only had to choose among alternative parts of speech for expressing a node; and Mann's PENMAN /Mann 1983/ really only had to order words and choose syntactic options.

This section briefly discusses some issues in designing a generator that handles all the complexities of word choice.

Design Issue 1: In what order are the factors considered?

As shown above, many factors can affect the decision to use a word. There are several ways to organize the factors.

Goldman's BABEL has tests organized into a discrimination network. This means it always performs tests in the same order. For example, given a conceptualization which includes INGEST, it always tests "is the object a medicine" before testing "is the object a liquid."

Another way to organize word choice is with a two-stage algorithm. For example, BABEL selects a primitive then discriminates; PAULINE gathers candidates then filters them for relevance; KING chooses associations to find a node then chooses among words for that node; and Thompson's model considers speaker's goals to produce an "intention" then consults syntax.

In FIG all factors contribute simultaneously.

Design Issue 2: Are all words chosen in the same way?

Many generators choose different types of words differently. Commonly distinguished are open-class words and closed-class words /Pustejovsky and Nirenburg 1987/ or content words and function words /Kempen and Hoenkamp 1987/, phrase-heads and modifiers /Goldman 1974/, and words with valence and words without.

FIG has one uniform process for all types of word choice. Everything which affects word choice is just a source of energy. Of course it is true that different types of factors are more important for different types of choices. For example, energy from the nodes of the input is typically more important for open-class words than for closed-class words. However, this fact does not affect the structure of FIG.

6. Design Principles

FIG addresses all the above issues in word choice. It works, not because of the details of representation and energy flow, but because it embodies several design principles. This section states these principles as general maxims for generators design.

Design Principle 1: Have an explicit representation of the status of the generation process at each point in time.

FIG has a complete and explicit representation of the state, syntactic and semantic, at each moment of the generation process. This representation consists of the activation levels of many concepts, syntactic constructions, and words. This represents which factors and choices are relevant; in other words, it constitutes the "working memory" of the generator. This representation makes all relevant information available for each successive decision to use a word.

This contrasts with generators in which information is implicit, for example, in the current value of a pointer or in the parameters of a function call. This also contrasts with generation based on stages. A stage model partitions the factors in choice into sets. There is no clear motivation for such a partition. Moreover, use of a stage model limits the availability of different types of information to different times.

Design Principle 2: Use a single, unified representation.

FIG is "unified" in two senses: all knowledge is part of one network, and information propagates freely by means of spreading activation. Nodes for compatible choices, of all sorts, are linked and therefore mutually reinforcing. This implies that activation levels tend to converge (or "settle" or "relax") into a state which represents a consistent set of choices.

This contrasts with modular generators. Modularity is surprisingly pervasive. Even generators which are unified in some respects are modular in other respects. The generators with uniform processing approaches, including Appelt's planning generator /Appelt 1985/ and Kalita's connectionist generator /Kalita and Shastri 1987/, employed levels of representation. Jacobs' KING exploited a uniform representation but relied on diverse algorithms and processes /Jacobs 1985b/. In addition, most generators partition knowledge into separate knowledge bases for dictionary, world knowledge, grammar rules.

The problem with modular design is that it does not support the flow of information *between* modules. This makes it hard for them to handle interactions between factors of different types. For example, the distinction between strategy and tactics requires an interface protocol between the two modules. This interface usually consists of a description of the information passed between the two. This information is variously called a "message," "meaning," "content," or "realization specification." Many have pointed out, however, that such a "message" can not contain enough information /Appelt 1985/ /Danlos 1984/ /Hovy 1987/. In particular, even seemingly mundane choices of words can be sensitive to the speakers goals. The underlying problem is that researchers have partitioned the *problem* in order to study it, which is reasonable; but they have also imposed partitions on the *designs for generators*, which is unjustified.

Of course it is impossible to *prove* that modular designs are inadequate. They can always be augmented with special pathways and protocols for the flow of information among modules. However, it is not obvious that patchwork design is unavoidable.

Design Principle 3: Do not rely on the details of the structure of the input.

The input to FIG is a structure of linked, activated nodes. These nodes are the ultimate source of energy that drives the entire generation process. However, there is no simplistic correspondence between input and output. This contrasts with generators which are designed around a well-elaborated notion of the input.

Most generators use inputs which are tailored to make generation easy, which means that they cannot handle inputs which are not "suitable." This constrains the concepts of the input to correspond to words of English in some fairly direct way. It also constrains the structure of the input to reflect the structure of English. It may constrain the input in other ways, for example, requiring the input to have a distinguished "top" node.

In contrast, FIG is free of the usual constraints on its input. FIG can easily emit words which are not directly related to the input, since choices are determined by spreading activation, which can come from diverse sources and follow long paths. Also, FIG builds up the structure of the output incrementally as a side effect of emitting words.

The only constraint that FIG imposes is that the input support activation flow. Thus it is *flexible* in that it can handle a wide variety of inputs. This contrasts with the usual practice of fixing an input format and insisting that anyone desiring to use the generator conform or write a pre-processor. The advantages of flexible generation for machine translation are obvious.

Design Principle 4: Treat most choices as emergent.

FIG does not explicitly "choose" concepts or syntactic structures. Such choices are unnecessary. The only explicit choices needed are the successive choices of words.

The appearance of syntactic choice emerges from the fact that constructions affect the form of the utterance. An analyst can, of course, look at an utterance and think "this exhibits the choice of construction X." However, FIG never actually explicitly chose X (although the node for X was probably highly activated and played an important role in the flow of activation). This contrasts with generators which explicitly make syntactic decisions, such as which template to use, which edge to traverse, how to order words, or whether to include or omit an optional constituent.

The appearance of concept choice emerges from the fact that words are associated with concepts, and so a word choice can imply the choice of a concept.

Choice *among* words is also emergent in FIG. For example, it never chooses between *"a"* and *"the."* The fact that *"a"* and *"the"* are in complementary distribution in English is represented with an inhibit-link between the nodes **'"a"** and **'"the"**. Thus, whenever one of these is activated the other receives negative energy. When generating, therefore, the network tends to settle into a state where one, but not both, of these nodes is highly activated. And thus typically only one of these words is selected. This is how FIG "chooses" between *"a"* and *"the,"* without treating them as explicit alternatives.

The problem with explicit choices is ordering them. It is hard, if not impossible, to fix an order such that no choice is made before a choice which it depends on /Danlos 1984/.

At this point I should acknowledge how subversive this approach really is. My guiding principle has been "word choice suffices." Intuitively, if every word choice is appropriate, then the whole utterance will be appropriate, by induction. Therefore it seems reasonable to study syntax and meaning in generation by focusing on the ways they affect word choice.

This contrasts starkly with most generation research, which seems to assume that "syntax constrains the problem of generation so well that word choice should be treated as an afterthought." In particular, the principle of emergent choice allow one to dispense with some things that generators are usually supposed to do. First, FIG does not produce a parse tree for a sentence while generating. I prefer to think of constructions existing in the generator during the production of a sentence rather than existing in the resulting utterance. In FIG many constructions are simultaneously active during production, with no mechanism other than spreading activation to unify or coordinate them. Second, FIG is not guaranteed to produce only grammatical utterances. I contend that grammaticality has been overemphasized. Output which is grammatically correct is not necessarily more understandable than fragmented, ungrammatical output.

7. Conclusions

Word choice involves a great deal of complexity. A spreading activation based design can handle the complexity, and produce high quality output. Designs based on the above principles also seem useful for cognitive modeling, since incremental generators can be used to model the pauses and errors of human speech performance.

References

Appelt, Douglas E., TELEGRAM: A Grammar Formalism for Language Planning, *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, (1983), pp. 595-599.

Appelt, Douglas E., *Planning English Sentences*, Cambridge University Press, 1985.

Chafe, Wallace L., The Deployment of Consciousness in the Production of a Narrative, in *The Pear Stories*, Wallace L. Chafe (editor), Ablex.

Danlos, Laurence, Conceptual and Linguistic Decisions in Generation, in *Proceedings 22nd Association for Computational Linguistics*, 1984, pp. 501-4.

Danlos, Laurence, *The Linguistic Basis of Text Generation*, Cambridge University Press, 1987.

Fillmore, Charles J., Frames and the Semantics of Understanding, *Quaderni di Semantica* VI(2) (1985).

Fillmore, Charles J., On Grammatical Constructions, textbook in preparation, University of California, Berkeley; Department of Linguistics, Fall 1987.

Fillmore, Charles J., Kay, Paul and O'Connor, M. C., Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone, *Language* (to appear).

Goldman, N., Computer Generation of Natural Language from a Deep Conceptual Base, Ph.D. Thesis, also Technical Report CS-74-461, Stanford University, 1974.

Hovy, Eduard, Generating Natural Language Under Pragmatic Constraints, CSD/RR #521, Yale, 1987.

Jacobs, Paul S., PHRED: A generator for natural language interfaces, Report CSD 85/198, University of California, Berkeley, 1985. revised version appears in *Computational Linguistics*, v.11, 1985.

Jacobs, Paul S., A Knowledge-Based Approach to Language Generation, PhD Thesis, also Report CSD 86/254, University of California, Berkeley, 1985.

Kalita, Jugal and Shastri, Lokendra, Generation of Simple Sentences in English Using the Connectionist Model of Computation, in *Ninth Cognitive Science Conference*, 1987, pp. 555-565.

Kempen, Gerard and Hoenkamp, Edward, An Incremental Procedural Grammar for Sentence Formulation, *Cognitive Science* **11**, 201-258 (1987).

Lakoff, George, *Women, Fire, and Dangerous Things*, University of Chicago Press, 1987.

Mann, William C., An Overview of the Penman Text Generation System, *Proceedings of American Association for Artificial Intelligence-83*, Washington, D.C., (1983), pp. 261-265.

McDonald, David D., Natural Language Generation as a Computational Problem, in *Computational Theories of Discourse*, Michael Brady (editor), MIT Press, 1983.

Norvig, Peter, A Unified Theory of Inference for Text Understanding, Report CSD 87/339, University of California, Berkeley, 1987.

Pustejovsky, James and Nirenburg, Sergei, Lexical Selection in the Process of Language Generation, in *Proceedings 25th Association for Computational Linguistics*, 1987, pp. 201-207.

Talmy, Leonard, Semantics and Syntax of Motion, in *Syntax and Semantics*, vol. 4, John P. Kimball (editor), Academic Press, New York, 1975.

Talmy, Leonard, Lexicalization Patterns: Semantic Structure in Lexical Forms, in *Language Typology and Syntactic Description, Vol. 3*, Timothy Shopen (editor), Cambridge, 1985. also UC Berkeley Cognitive Science Program Report No 30.

Thompson, Henry, Strategy and Tactics: A Model for Language Production, *Proceedings of the Chicago Linguistics Society*, Chicago, (1977) vol. 13.

Wilensky, Robert, Some Problems and Proposals for Knowledge Representation, Report CSD 87/351, University of California, Berkeley, 1987.