SRI International

SPEECH GENERATION FROM SEMANTIC NETS

Technical Note 115

September 1975

By: Jonathan Slocum
Artificial Intelligence Center

SRI Project 3804

Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston Massachusetts, October 30 - November 1, 1975.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under Contract No. DAHC04-75-C-0006.

SRI International
333 Ravenswood Ave. • Menlo Park, California 94025
(415) 326-6200 • Cable: SRI INTL MPK • TWX: 910-373-1246

SPEECH GENERATION FROM SEMANTIC NETS

SUMMARY

Natural language output can be generated from semantic nets by processing templates associated with concepts in the net. A set of verb templates is being derived from a study of the surface syntax of some 3000 English verbs: the active forms of the verbs have been classified according to subject, object(s), and complement(s); these syntactic patterns, augmented with case names, are used as a grammar to control the generation of text. This text in turn is passed through a speech synthesis program and output by a VOTRAX speech synthesizer. This analysis should ultimately benefit systems attempting to understand English input by providing surface structure to deep case structure maps using the same templates as employed by the generator.

INTRODUCTION

If computers are to communicate effectively with people, they must speak, or at least write, the user's natural language. The bulk of the work in computational linguistics has been devoted to computer understanding of natural language input, but relatively little effort has been expended in developing natural language output. Most English output systems have been along the line of "fill in the blank" with perhaps some semantic constraints imposed; there have been few attempts at language generation from what one could call "semantic net" structures (Simmons and Slocum, 1972; Slocum, 1973; Goldman, 1974).

Perhaps generation is considered a much easier problem. The success of understanding efforts is generally believed to depend on some workable theory of "discourse organization" which would account for effects of context and would show how anaphoric expressions (pronouns and noun phrases) are resolved and how sentences are ordered in the output. As it happens, these mechanisms are precisely those that a "response generator" must incorporate if it is to appear intelligent. The study of generation will play an important role in solving the problem of understanding if it can demonstrate a mapping from deep semantic structures to surface strings.

Let us briefly outline some relevant processes in the speech understanding system being developed by SRI and SDC (Walker et al., 1975, and Ritea, 1975). The user initiates a session by establishing communication with the system; all subsequent dialog

(input and output) is monitored by a "discourse module" (Deutsch, 1975) to maintain an accurate conversational context. An executive coordinates various knowledge sources -- acoustic, prosodic, syntactic, semantic, pragmatic, and discourse -- to "understand" successive utterances.

The analyzed utterance is then passed to the "responder" -- another component of the discourse module. The responder may call the question-answerer if the input is a question; it may call a data base update program if the input is a statement of fact; or it may decide on some other appropriate reply. The content of the response is passed to the generator, perhaps with some indication of how it is to be formulated. The reply may be a stereotyped response ("yes", "no", "I see"), a noun phrase (node), a sentence (verb node), or, eventually, a paragraph.

The generator outputs stereotyped responses immediately; if the response is more complicated (a "noun" node, "verb" node, or eventually a network), a more detailed program is required. This program will determine exactly how the response is to be formulated -- as an NP, S, or sequence of Ss; it may be required to choose verbs and nouns with which to express the deep case net structures, as well as a syntactic frame for the generation. The generator produces the response in "text" form; this in turn is passed to a speech synthesis program for transformation and output by a commercial VOTRAX speech synthesizer. Currently no sentence intonation or stress contouring is being performed. Since the major interest of this paper is in "text" generation,

no further reference to the synthesis step will be made.

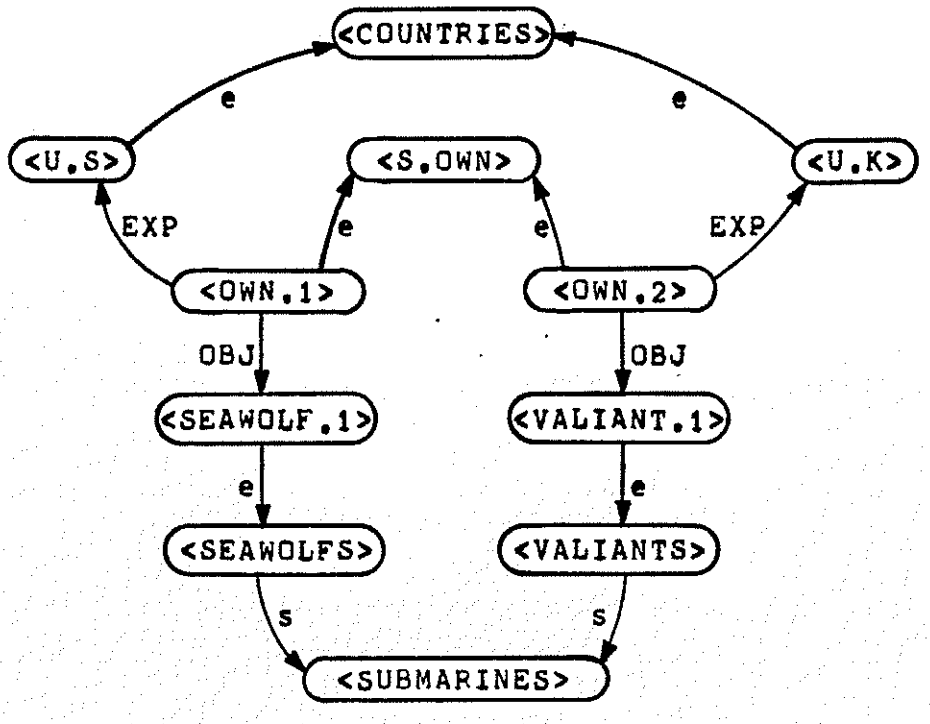
CONSTRAINTS ON RESPONDING

There are several considerations involved in responding appropriately to an utterance. First, there are "conversational postulates" (Gordon and Lakoff, 1975) shared by the users of a language; these postulates serve to constrain the content and form of communications from the speaker to the hearer. For instance, the speaker should not tell the hearer something the hearer already knows, lest he be bored; yet the speaker cannot tell the hearer something the hearer knows absolutely nothing about, or the hearer will not comprehend. The speaker should relate the news in his message to the prior knowledge of the hearer; this requires the speaker to have a model of the hearer. These heuristics must operate in conjunction with a "response producer" to constrain what may be output by a "sentence" generator. We are only beginning to understand how to incorporate these postulates in a language processing system.

Then there is the matter of constructing the basic sentence. Normal English syntax requires at least one verb in the sentence; choosing a main verb constrains the surface structure. For instance, in the absence of compounds any verbs other than the main verb will have to appear in another form: nominal, infinitive, gerund, participle, or subordinate clause. How does the relevant information contained in a semantic net indicate the appropriate form? The traditional answer is "by means of the lexicon." We will explore the relationship between net and

lexicon and advance a methodology for representing a map from deep case structure to surface structure.

This paper focuses on a philosophy of single-sentence formatting: choosing a main verb, choosing the gross structure of the output sentence, and deciding how to generate appropriate noun phrases. Our examples will employ simplified semantic net structures, somewhat like those in the actual SRI "partitioned semantic net" system (Hendrix, 1975). Nodes in the net may represent physical objects, relationships, events, sets, rules, or utterances, as in the example below. Directed labelled arcs connect nodes and represent certain "primitive" time-invariant relationships.



In the net fragment above, the U.S. and the U.K. are elements (e) of the set of countries. As EXPeriencers they each participate

in OWNing situations involving as OBJects particular submarines; each submarine is an element of some class of submarines, and these classes are subsets (s) of the set of all submarines.

GENERATION TEMPLATES

The first requirement for generation is to derive some templates for English sentences. We choose a simple verb for demonstration -- OWN. We note that our verb has several "synonyms": HAVE, POSSESS, and BELONG (TO). Since each of these verbs (including OWN) has other sense meanings, we posit a node <S.OWN> in the net that corresponds to the abstract "ownership" sense they have in common; this node will be the "prototypical" OWN, in that it will incorporate the "meaning" of the situation of owning (including any semantic constraints on its arguments), and in that all instances of owning situations will be related to it. With this node we will associate the appropriate verbs (OWN, POSSESS, HAVE, BELONG) and templates. Note that one template will not suffice for all four verbs; for instance, the subject of BELONG is the OBJect entity, while in the other (active) verbs the subject is the EXPeriencer:

```
EXP owns OBJ ; OBJ is owned by EXP
EXP possesses OBJ ; OBJ is possessed by EXP
EXP has OBJ ; OBJ belongs to EXP
```

So we propose the corresponding templates:

```
[OWN (EXP Vact OBJ) (OBJ Vpas BY EXP)]
[POSSESS (EXP Vact OBJ) (OBJ Vpas BY EXP)]
[HAVE (EXP Vact OBJ)] [BELONG (OBJ Vact TO EXP)]
```

Now, in order to speak about a particular owning situation, we pursue the hierarchy to find the "canonical" S.OWN, choose a verb

(say, BELONG) and an associated template (OBJ Vact TO EXP), and generate the constituents consecutively.

But we have a problem; there is no indication of how the EXP and OBJ arguments are to be generated. NP will not always suffice; note for instance that the predicate argument of "hope" in "John hoped to go home" must be an infinitive phrase (rather than the gerund phrase that NP might produce). Even a cursory study of a few hundred verbs in the language shows that they have very definite (and regular) constraints on the syntactic form of their constituents. These constraints appear to be matters for the lexicon rather than the grammar. Therefore, we associate verbs and templates with word senses (prototypical nodes in the net) rather than implement them via grammar rules, and we explicitly incorporate the constituent types in the templates:

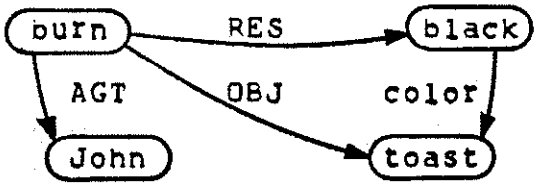
```
[OWN ((NP EXP) Vact (NP OBJ)) ((NP OBJ) Vpas BY (NP EXP))]
[POSSESS ((NP EXP) Vact (NP OBJ)) ((NP OBJ) Vpas BY (NP EXP))]
[HAVE ((NP EXP) Vact (NP OBJ))]
[BELONG ((NP OBJ) Vact TO (NP EXP))]
```

A set of patterns like these is associated with every "prototype verb" node in the knowledge base. It would seem that all we need is an interpreter that, given any "verb instance" node in the knowledge base, looks up the patterns for that type of node, chooses a verb, a corresponding template for the verb, and then proceeds to "evaluate" the pattern:

```
verb [OWN.1-->S.OWN] --> belong
temp --> [(NP OBJ) Vact TO (NP EXP)]

(NP OBJ) --> the SeaWolf
Vact --> belongs
TO --> to
(NP EXP) --> the U.S.
```

But we still run into trouble with our simple scheme. Consider the sentence, "John burned the toast black."



By using the simple pattern ((NP AGT) Vact (NP OBJ)) we could easily generate the "incorrect" sentence, "John burned the black toast," since (NP OBJ) might include the color of the toast. We need a pattern more like ((NP AGT) Vact (NP OBJ) (Mod RES)), in which the RESult of the action will be directly related to the verb. However, this is not quite enough -- at least, not without a very complicated interpreter -- because the interpreter must know that (NP OBJ) cannot include the verb's RES argument (black). Thus, by convention, we may indicate an extra argument to be passed to a constituent generator (such as the function NP) to denote the item(s) not to appear in the resultant constituent:

((NP AGT) Vact (NP OBJ RES) (Mod RES))

The pattern (NP OBJ RES) means "generate an NP using the OBJect of the verb, but do not include the RESult of the verb in the NP." This convention actually prevents enormous proliferation of patterns (i.e., a pattern copy for every possible "missing" constituent). This level of detail would be unreasonable if few other verbs could use this template; however, there are more than a hundred verbs that share this same pattern. Since there are relatively few templates, each shared by several tens or hundreds of verbs, the use of templates proves to be quite helpful.

There are other sources of potential pattern proliferation, an important one being the combinatorial arrangements of case arguments of time, manner, and other such adverbials, as well as other (possibly non-adverbial) case arguments such as source, goal, instrument, etc. Some of these arguments are rather constrained in their positions in the sentence, but others may appear almost anywhere:

"Yesterday the ship sailed from the lighthouse to the dock."
"The ship sailed from the lighthouse to the dock yesterday."
"Yesterday the ship sailed to the dock from the lighthouse."

It is of course unreasonable to try to maintain all the possible patterns; instead we leave insertion of these adverbial arguments to a single heuristic routine (described below). There are several justifications for this, among them: (1) the particular form of the verb cannot be generated until the subject, object(s) and complement(s) have been generated, (2) these adverbials are so universal as to appear in almost any of the patterns and in several possible places, and (3) there are some heuristic constraints involved in the placement of arguments.

One may question whether passive templates should be stored; certainly, they could be derived. On the other hand, neglecting to store them would force us to indicate with each verb (sense), whether it can (or, sometimes, must) be passivized. Indicating "transitive" is not enough since there are transitive verbs (i.e., verbs that take an object) that cannot be passivized. Since we have to store the information anyway, we can save some code and computing time by storing the passive template.

There are several reasons for generating the verb after the major arguments. First the subject must be generated so that the verb can be made to agree in number. Second, certain word senses are true of verb-particle combinations while not of the isolated verb. Since, in addition, particles must appear after objects that are short (like pronouns) but before objects that are long (like noun phrases), the particle must be positioned after the object is generated. Finally, insertion of some adverbials (e.g. "not") requires an auxiliary verb -- thus verb generation must follow adverbial generation.

VERB PATTERNS

This study started with the 25 "verb patterns" presented by Hornby (1954). These in turn came from a dictionary by Hornby et al. (1948). Verbs in the dictionary are classified according to their gross syntactic patterns of subject, object(s), and complement(s); most of the patterns are sub-divided. The authors claim that these patterns account for all constructions involving all the verbs in their dictionary -- and, by extension, in the language. This classification is not immediately useful to computational linguists since it does not address underlying semantics. Nevertheless, it is clear that it can serve as the basis for a derivation of underlying case structures and, particularly, as a basis for "generation templates."

These patterns are being converted into templates much like those derived earlier; the analysis is being performed with respect to about 3000 verbs drawn from the dictionary (Slocum, to

appear). These templates serve as the major portion of a modular "generation grammar," with the remainder in the form of heuristic functions for constructing syntactic constituents.

NOUN PHRASES

What to include in a noun phrase should be another matter for the discourse module to judge. There are no well-formulated rules accounting for anaphora in English; indeed, there are few well-established parameters other than that the hearer must be able to resolve the (pro)nouns to their referents. The speaker should employ anaphora in order to avoid repetition, but only if his model of the hearer indicates that the hearer can resolve the ambiguity. There are some low-power pronominalization rules that could be directly incorporated in a generator -- reflexivization, for example. Nevertheless, it is important to realize that when a generator is unaware of the conversational context, it should not independently decide how to generate noun phrases; it can only decide when to do so. This situation has not been universally recognized, but it is becoming increasingly clear that a discourse module must be consulted during the generation phase. The discourse module will not know ahead of time what NPs are to be produced unless it performs many of the same operations that the generator would do anyway. Yet the context-sensitive decision strategy may have to resort to such measures as disambiguating the proposed output using the model of the hearer in order to determine what anaphora is resolvable. It is unreasonable to incorporate this strategy in the generator, since

for many reasons it must be part of the discourse module.

Therefore the generator should pass any "noun" constituent to the discourse module (perhaps with its recommendation about how to produce the constituent); the module must determine if a pronoun or bare noun is ambiguous to the hearer, and, if so, what to add to the noun in order to make the desired referent clear. In the current SRI system, noun patterns (Slocum, to appear) are used to control noun phrase generation. Much like verb patterns, noun patterns order the constituents in the phrase and indicate how each constituent is to be generated by naming a function to be called with the network constituent:

[(DET) (Adj QUAL) (Adj SIZE) (Adj SHAPE) (Adj COLOR) (N)]

Patterns like this are distributed about the network hierarchy; in the future, the discourse module will decide for each pattern constituent whether it is to appear in the phrase.

HEURISTIC RULES

Hornby describes three basic positions for adverbs in the clause: "front" position, "mid" position, and "end" position. Front position adverbs occur before the subject: "Yesterday he went home; from there he took a taxi." The interrogative adverbs (e.g. how, when) are typically constrained to front position; others may appear there for purposes of emphasis or contrast.

Mid position adverbs occur with the verb (string); if there are modal or auxiliary verbs, the adverb occurs after the first one. Otherwise the adverb will appear before the verb, except for "unstressed" finites of be, have, and do: "we often go

there"; "she is typically busy"; "he is still waiting."

End position adverbs occur after the verb and after any direct or indirect object present. While relatively few clauses have more than one adverb in front position or more than one in mid position, it is common for several adverbs to appear in end position in the same clause: "they play the piano poorly together".

Adverbials of time (answering the question, "when?") usually occur in end position, but may appear in front position for emphasis or contrast. Adverbials of frequency (answering the question, "how often?") can be split into two groups. The first group is composed of single-word adverbs that typically occur in mid position but also may be in end position; the second is composed of multiple-word phrases that appear in end position or, less frequently, in front position. Adverbs of duration ("[for] how long?") usually have end position, with front position for emphasis or contrast. Adverbs of place and direction normally have end position. Adverbs of degree and manner have mid or end position, depending on the adverb.

Along with such rules concerning the positions of various types of adverbs, there must be a mechanism to order the adverbs that are to occur in the "same" position. There are some heuristics: among adverbials of time (or place) the smaller unit is usually placed first, unless it is added as an afterthought: "the army attacked the village in force on a hot August afternoon, just after siesta". Adverbials of place and direction

usually precede those of frequency, which in turn precede those of time.

These rules are implemented in the same routine that produces the verb; when a template is first interpreted -- much as a sequence of function calls -- the "Vact" or "Vpas" keys are ignored. Once the subject, object(s) and complement(s) indicated by the template are generated, this "clean up" routine is called. It employs the heuristics described above to add the adverbial constituents and verb, then concatenates the constituents to produce a complete clause.

DISCUSSION

In theory, the set of possible English sentences is infinite. The obvious question then arises, "If one tries to account for them with templates, won't there be an infinite number of templates?" The simple answer is, "No, for some of the same reasons that allow a finite grammar to generate an infinite number of strings." One can produce sentences of arbitrary length by (1) arbitrary embedding, and (2) arbitrary conjunction. One does not do so by including arbitrary numbers of distinct case arguments. Even so the number of basic patterns could be extremely large. Evidence, however, is to the contrary: the eventual number of templates would appear to be several times the number of patterns, owing to the substitution of particular prepositions for "prep" in the syntactic patterns, and the assignment of different case names to a particular constituent depending on the particular verb used.

REFERENCES

Deutsch, Barbara G. Establishing Context in Task-Oriented Dialogs. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Goldman, Neil M. Computer Generation of Natural Language from a Deep Conceptual Base. AI Memo 247, Artificial Intelligence Laboratory, Stanford University, Stanford, California, 1974.

Gordon, David, and Lakoff, George. Conversational Postulates. Syntax and Semantics, Volume 3: Speech Acts. Edited by Peter Cole and Jerry L. Morgan. Academic Press, New York, 1975.

Hendrix, Gary G. Expanding the Utility of Semantic Networks through Partitioning. Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, 3-8 September 1975, 115-121.

Hornby, A. S., Gatenby, E. V.,and Wakefield, H. The Advanced Learner's Dictionary of Current English. Oxford Press, London, 1948.

Hornby, A. S. A Guide to Patterns and Usage in English. Oxford Press, London, 1954.

Ritea, H. Barry. Automatic Speech Understanding Systems. Proceedings, Eleventh Annual IEEE Computer Society Conference, Washington, D. C., 9-11 September 1975.

Simmons, Robert F., and Slocum, Jonathan. Generating English Discourse from Semantic Networks. Communications of the ACM, 1972, 15, 891-905.

Slocum, Jonathan. Question Answering via Canonical Verbs and Semantic Models: Generating English from the Model. Technical Report NL-13, Department of Computer Sciences, University of Texas, Austin, Texas, January 1973.

Slocum, Jonathan. Verb Patterns and Noun Patterns in English: A Case Analysis. Artificial Intelligence Center, SRI, Menlo Park, California, (in preparation).

Walker, Donald, E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.