GEMS: A MODEL OF SENTENCE PRODUCTION

Domenico Parisi Alessandra Giorgi
Istituto di Psicologia del C.N.R.
Reparto Processi Cognitivi e Intelligenza Artificiale
Via dei Monti Tiburtini, 509
00157 Roma, Italy

ABSTRACT

The paper describes GEMS, a system for Generating and Expressing the Meaning of Sentences, focussing on the generation task, i.e. how GEMS extracts a set of propositional units from a knowledge store that can be expressed with a well-formed sentence in a target language. GEMS is lexically distributed. After a central processor has selected the first unit(s) from the knowledge store and activated the corresponding lexical entry, the further construction of the sentences meaning is entrusted to the entries in the vocabulary. Examples of how GEMS constructs the meaning of a number of English sentence types are briefly described.

1. Constructing the meaning of sentences

Most work on natural language generation has been concerned with the production of connected text (Davey, 1979; Goldman, 1975; Mann and Moore, 1981; Meehan, 1977) or with language generation as a goal-directed, planned activity (Appelt, 1980; Mann and Moore, 1981). Less attention has been dedicated to the linguistic details of sentence generation, i.e. to constructing a general device for imposing the appropriate linguistic form to the content that must be expressed (but see Kempen and Hoenkamp, 1982).

The aim of this paper is to describe GEMS, a system for Generating and Expressing the Meaning of Sentences. GEMS takes a store of knowledge as input and gives English sentences expressing that knowledge as output. The knowledge contained in the knowledge store is purely conceptual knowledge with no trace of linguistic form. There is no partitioning of knowledge in parts which can be expressed by single sentences or by single lexical items, no grammatical labelling of items as verbs, nouns, or subjects, objects, etc., no other traces of syntactic or lexical form. Hence, a first task of GEMS is to extract from the knowledge store the knowledge which it is appropriate to express in a well-formed sentence, i.e. to generate the meaning of the sentence. Since the meaning thus constructed must be expressed with a specific sequence of words, two further tasks of GEMS are to select the semantic and grammatical morphemes that make up the sentence and to put them in the appropriate sequential order.

Producing sentences is a goal-directed activity: what one says depends on one's goals. GEMS however is a model of how to say something, not of what to say. When it arrives at a decision point on what to say, GEMS makes a random choice. Hence, GEMS is not a complete model of the activity of producing sentences but only a model of the linguistic constraints on the communication of knowledge and ideas.

GEMS conceives the knowledge necessary to produce sentences as largely distributed in the lexicon. This change from previous more centralized version of GEMS (see Parisi and Giorgi, 1981; 1983) has been suggested to us by Oliviero Stock and Cristiano Castelfranchi and it is related to our view of a lexically distributed sentence comprehension process (see Stock, Castelfranchi, and Parisi, 1983; Parisi, Castelfranchi, and Stock, in preparation). The lexical entries are procedures that activate each other in a given order when a sentence is produced, although the order of activation may not coincide with the external sequential order of the words in the actual sentence. When executed the entries' procedures (a) extract the sentence's meaning from the knowledge store, (b) lexicalize this meaning with the appropriate semantic and grammatical morphemes, and (c) put these morphemes in the correct sequential order. A central processor has the task of searching the knowledge store for knowledge to be expressed and the lexicon for the lexical entries that can express this knowledge. However, the main task of the central processor is to start the construction process and to keep a record of the order of activation of the lexical entries. The overall scheme of GEMS is represented in Fig.1

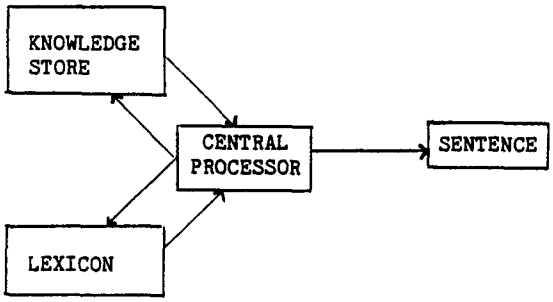

Fig.1. Overall scheme of GEMS

In the present paper our purpose is to describe GEMS with respect to its first task, i.e. how GEMS generates the meanings of sentences by extracting syntactically appropriate knowledge from the knowledge store. We will proceed by first describing the knowledge store, the vocabulary, and the central processor, and then briefly analyzing some sentence types to show how GEMS constructs their meanings.

2. The knowledge store

The world knowledge of the system, or as we will say, its encyclopedia (ENC), is represented as a set of propositional units. A propositional unit is made up of a predicate, the predicate's arguments, and a label that uniquely identifies each unit. Argument and labels have number codes that indicate when they refer to the same entity (same code) or to different entities (different codes). Labels are represented as Cs whereas arguments can be either Xs or Cs. When an argument of a unit is a C, this means that the unit is a "recursive" one, i.e. a unit which takes another unit as its argument. In such case the C argument is the label of the unit taken as an argument.

Let us assume that the system has the knowledge items represented in (1), i.e. (1) is the system's ENC. Obviously, neither the absolute numbers assigned to the arguments and labels nor the order of listing of the units in (1) have any meaning.

(1)
C1: X1 BILL	C6: X1 THINK C7
C2: X1 SEE X2	C7: X2 LEAVE
C3: X2 MARY	C8: X4 DOG
C4: X3 ARRIVE	C9: X4 SLEEP
C5: X3 JOHN	C10: C9 DEEP

As (1) makes it clear, no traces of linguistic form are present in ENC. The knowledge items in (1) are not marked as being nouns, verbs, or any other grammatical classes; furthermore, nothing is subject, object, attribute, or any other functional class. Finally, there is no indication in (1) of which items make up a well-formed sentence or other syntactic phrases.

3. The lexicon

In order to extract a syntactically well-formed meaning from ENC and express it with the appropriate sequence of semantic and grammatical morphemes the system utilizes a vocabulary (VOC). VOC is a set of meaning/signal pairs called lexical entries. GEMS' vocabulary is a morphological one, i.e. the vocabulary includes lexical entries which are "roots" (e.g. see-) and lexical entries which are "(inflexional) suffixes" (e.g. -s). However, for the purpose of describing the sentence meaning construction process we can assume a simplified vocabulary of whole words.

The meaning of a lexical entry is made up of four components.

(a) There are first of all one or more propositional units with the same types of predicates that are found in ENC. The only difference is that the units which are found in a lexical entry have letter codes and not number codes on their arguments and labels. (The number codes show the linkings among the various units within ENC and, as we will see, within a sentence's meaning. The letter codes indicate the linkings among units within a single lexical entry.) These propositional units represent the semantic content of a lexical entry. They are called semantic units (SU). Even though the SUs of an entry may be more than one, we will represent the semantic content of the entries with a single SU, i.e. without lexical decomposition.

(b) Secondly, the meaning of a lexical entry contains a list of one or more "saturation instructions" on the arguments of the SUs. These saturation instructions correspond to the assembly instructions that play a central role in the sentence comprehension process (see Stock, Castelfranchi, and Parisi, 1983), where they serve to assemble together in the appropriate way the separate meanings of the words making up the sentence to be understood. A saturation instruction is "on" a given argument of the SUs of the lexical entry. For example, a verb like to take has a SU "CA: XA TAKE XB" and two saturation instructions on XA and XB, respectively. A noun like president has a SU "CA: XA PRESIDENT XB" and a saturation instruction on XB. A saturation instruction on a given argument is a procedure for (i) extracting from the knowledge store a propositional unit having the argument to be saturated as its argument or its label, and (ii) identifying a lexical entry in VOC which has the extracted

propositional unit among its SUs.

(c) A third component of a lexical entry is a "marker". Lexical entries contain one of three types of markers: TEMP, HEAD, and ADV. TEMP is a marker of verbs (full verbs not copula or auxiliary verbs), adjectives and some uses of "semantic" prepositions (as in The book is for Susan, The bottle is on the table). HEAD is a marker of nouns (including nominalizations like arrival). ADV is a marker of adverbs, subordinating conjunctions, and some other uses of "semantic" prepositions (as in Bill is eating in the kitchen). Markers are procedures for selecting the next step to be taken by the meaning construction process when the saturation instructions of a lexical entries have all been executed. As procedures markers make reference to the record of the order of activation of the lexical entries which is kept by the central processor. Therefore, we will explain the meaning of TEMP, HEAD, and ADV after describing the central processor.

(d) Finally, lexical entries include as a fourth component one or more additional propositional units having special predicates which are different from the semantic predicates of the units in ENC and the SUs in the vocabulary entries. These special units control the lexicalization of the grammatical morphemes and therefore they won't be mentioned in this paper.

4. The central processor

The central processor executes the procedures of the lexical entries, both the saturation instructions and the markers. However in addition it has two specific tasks of its own which represent the non-lexically distributed portion of GEMS.

First of all, the central processor starts the whole process by selecting in ENC a unit having a specified argument as one of its arguments or as its label, and then looking up in VOC a lexical entry that can lexicalize this unit, i.e. that has this unit as the lexical entry's SU. This is the first step of the sentence production process and it is the central processor which is responsible for it.

Secondly, the central processor keeps a record of the order of activation in VOC of the lexical entries that will make up the sentence (more precisely, the sentence's "content words"). The meaning of the sentence to be produced is constructed step by step by activating and executing the meanings of these lexical entries. In order to control this process GEMS must rely on a trace of the path traversed by the lexical activation process. More specifically, for each lexical entry which is activated there is a record of the lexical entry that activated the entry. This allows the system at any time to "step back", i.e. to trace back from an active lexical entry to the lexical entry that activated it. The latter entry becomes the new active lexical entry.

We can now return to the markers contained in the lexical entries and explain the meaning of HEAD, TEMP, and ADV. As already noted, these are names of procedures that are executed after all the unsaturated arguments of the lexical entry have been saturated.

HEAD is a very simple instruction to step back to the lexical entry from which the system originally moved to the currently active lexical entry (ALE), and to make this entry the new ALE. As we know HEAD is carried by nouns and therefore it is an instruction to move from the current noun to the governing verb (Bill sleeps), noun (the president of the company) or adverbial preposition (in the garden).

TEMP is a two step procedure. The first step is a recursive instruction to search ENC for a unit which has the label of the current ALE as one of its argument and then lexicalize this unit. Since TEMP is carried by verbs and adjectives, it is an instruction for constructing one or more adverbials modifying the verb or adjective (Bill sleeps deeply, Mary is very nice, Bill sleeps deeply in the bed). When this first step has been executed TEMP has a second instruction to step back. This allows the system to step back from a subordinate clause verb to the governing verb, noun or adverbial conjunction (Bill thinks that Mary left, The announcement that Bill had won delighted Peter, When Bill went to New York Mary was relieved). If there are no entries to step back to, the construction process ends.

ADV is very similar to TEMP. It first attempts to construct recursive adverbials in ENC (adverbials modifying adverbials, e.g. Bill sleeps very deeply) and then it steps back, ultimately to the verb or adjective being modified.

Before proceeding to analyze how GEMS constructs the meaning of various English sentence types it is necessary to note two limitations of the system as it is now.

A first limitation is that the procedure produces sentences only in response to a question to say something on a specific entity that is pointed out to the system from outside. An example could be "Say something on Napoleon". The system's response would be to produce a sentence expressing some

knowledge it has about Napoleon. A second limitation is that GEMS does not produce sentences containing pronouns and sentences where the starting entity is not included in the sentence's main clause. An extension of GEMS to sentences containing pronouns is described in Giorgi and Parisi (1984). As for sentences with the starting entity outside their main clauses they raise problems related to the status of the propositional units in ENC, i.e. whether a particular unit is "believed" by the system or not (for a treatment within the present framework, see Castelfranchi, Parisi and Stock, 1984). If the starting entity is "Mary" and the system knows that Bill thinks that Mary left it would not be appropriate for the system to produce a statement like Mary left. However, we won't deal with these problems in the present paper.

5. How the meaning of various sentence types is constructed

Consider how the meaning of a simple sentence like Bill saw Mary is constructed by GEMS.

Let us assume that the system is asked to produce a sentence about Mary, or more precisely about argument X2 (see the encyclopedia in (1)). The central processor searches ENC for a unit having X2 as its argument or label. The unit "C2: X1 SEE X2" is selected. The central processor looks up in VOC a lexical entry having a corresponding unit among its SUs. Assume that VOC contains lexical entry (2).

(2) Semantic Units	Marker	Saturation Instruct.		
CA: XA SEE XB	TEMP(CA)	XA,XB	saw	32

This entry is identified and it becomes the "active lexical entry" (=ALE). Its identification number, 32, is recorded by the central processor along with the activating agent. Since the activating agent in this case is the central processor (CP) itself the pair "CP, 32" is recorded.

Now the meaning of the entry executes itself. Since the entry contains two saturation instructions they are executed in whatever order. Assume that XA is tackled first. The processor searches ENC for a unit having XA, or more precisely its corresponding argument in ENC, X1, as one of its arguments or as its label. The unit "C1: X1 BILL" is selected. To lexicalize this unit the processor identifies lexical entry 14 in VOC:

(3)

CA: XA BILL	HEAD(XA)	---	Bill	14

Entry 14 becomes the new ALE and the processor record its identification number, 14, along with the identification number of the activating entry: "32, 14".

The entry Bill has no saturation instructions. Therefore, the processor executes its marker: HEAD(XA). It steps back, i.e. it makes the activating entry, 32, the new ALE.

The new ALE, saw, has a further argument to be saturated: XB(=X2). This leads to the selection of unit "C3: X2 Mary" in ENC and to the identification of the following entry in VOC:

(4)

CA: XA MARY	HEAD(XA)	---	Mary	5

5 is the new ALE. The processor records "32, 5". Since Mary doesn't have saturation instructions, HEAD directs the system to step back to 32 again.

At this point there are no further instructions of saw and the entry's marker, TEMP(CA), can be executed. TEMP checks whether there are in ENC propositional units having CA(=C2) as their argument that the system may want to express (as averbials). Since the answer is No, TEMP directs the system to step back. But there is no lexical entry to step back to because saw is the initial lexical entry, i.e. the entry initially activated by the central processor. Hence, the meaning construction process ends here. The meaning of the sentence Bill saw Mary has been constructed.

The mechanism of the saturation instructions allows for an indefinite "going down" of the construction process. A noun phrase like the president of the company in the sentence Bill saw the president of the company is constructed by first selecting a noun which has a saturation instruction (president) and then a further noun to saturate that instruction (company). When company is reached, since this noun has no saturation instructions, the system steps back first to president and then to the initial verb saw.

In a similar way the meaning of nominalizations like John's arrival (see (1)) can be generated using the following lexical entry for arrival:

(5)

CA: XA ARRIVE	HEAD(CA)	XA	arrival	15

Subordinate clauses, i.e. verb-, noun-, and adverbial-complements, can all be generated by the same mechanism. The only difference is

that when their meaning has been completed the TEMP marker of the subordinate clause verb directs the system to step back to the higher verb, noun or adverbial to continue with the construction process at the higher level.

Consider how the meaning of the sentence Bill thinks that Mary left is constructed. Let us assume that the system is asked to produce a sentence about X1 (Bill) and that the unit which is selected in ENC is "C6: X1 THINK C7". This unit is lexicalized with the following entry:
(6)
CA: XA THINK CB TEMP(CA) XA, CB thinks 81

If the argument CB (=C7) is first taken up for saturation, this leads to the selection of unit "C7: X2 LEAVE" in ENC and the activation of the entry left in VOC. At this point left is the new ALE. Its only argument is saturated with Mary and then the system steps back first to left and then to thinks. Thinks has another argument to be saturated, XA (=X1). The system saturates X1 with Bill. Thus, the meaning of Bill thinks that Mary left has been completed.

Adverbials modifying verbs or adjectives are also generated by TEMP. Consider the sentence The dog sleeps deeply. When the saturation instruction of sleeps has been executed (thereby generating the meaning of dog), the TEMP marker of sleeps searches ENC for units having the TEMP-marked argument (C9) as one of their arguments. The unit "C10: C9 DEEP" is found. This unit is lexicalized with the entry:
(7)
CA: CB DEEP ADV(CB) --- deeply 36

Since deeply has no saturation instructions and its marker ADV cannot find further adverbials in ENC, the system steps back to saw and the construction process ends. The meaning of the sentence The dog sleeps deeply has been constructed.

GEMS can be slightly modified to generate equative sentences (Fido is a dog) and sentences containing noun modifiers (a nice girl, the girl who was smiling). Furthermore, GEMS can also deal with cases where the initial lexical entry activated by the central processor is not a TEMP-marked entry, as it was the case in the examples analyzed above, but it is a HEAD- or an ADV-marked entry, i.e. a noun or an adverb.

A version of GEMS for one-clause Italian sentences has been implemented by G.Adorni in FranzLisp on a VAX computer at the University of Genova.

"REFERENCES"

Appelt, D.E. Problem solving applied to language generation. Proceedings of the 18th Annual Meeting of ACL, 1980, pp.59-63.

Castelfranchi, C., Parisi, D., Stock, O. Extending the expressive power of proposition nodes. In B.G.Bara and G.Guida (eds.), Computational Models of Natural Language Processing. Amsterdam: North Holland, 1984.

Davey, A. Discourse Production. Edinburgh: University Press, 1979.

Giorgi, A., Parisi, D. Producing sentences containing pronouns. RPCIA/17, Istituto di Psicologia, CNR, 1984.

Goldman, N.M. Conceptual generation. In R.Schank (ed.), Conceptual Information Processing. Amsterdam: North Holland, 1975.

Kempen, G., Hoenkamp, E. An incremental procedural grammar for sentence formulation. Unpubliched paper, University of Nijmegen, 1982.

Mann, W.C., Moore, J.A. Computer generation of multiparagraph English text. American Journal of Computational Linguistics, 1981, 7, 17-29.

Meehan, J.R. Tale-spin, an interactive program that writes stories. Proceedings of the 5th IJCAI, 1977, pp.91-98.

Parisi, D., Giorgi, A. A procedure for the production of sentences. RPCIA/1, Istituto di Psicologia, CNR, 1981.

Parisi, D., Giorgi, A. A procedure for constructing the meaning of sentences. RPCIA/7, Istituto di Psicologia, CNR, 1983.

Parisi, D., Castelfranchi, C., Stock, O. A model of sentence comprehension and production, in preparation.

Stock, O., Castelfranchi, C., Parisi, D. WEDNESDAY: Parsing flexible word order languages. Proceedings of the 1st Meeting of ACL, European Chapter, 1983.