INTERFACE. Journal of Applied Linguistics 3.1 (1988), 33-44

THE SEMSYN GENERATION SYSTEM : INGREDIENTS, APPLICATIONS, PROSPECTS

Dietmar RÖSNER
Universität Stuttgart

ABSTRACT

We report about the current status of the SEMSYN generation system. This system -- initially implemented within a Japanese to German MT project – has been applied to a variety of generation tasks both within MT and text generation. We will work out how these applications enhanced the system's capacities.

1. THE STARTING POINT

The SEMSYN project began in 1983 with an MT application as starting point. We had to investigate the possibility of using semantic structures derived from Japanese as input to a generator for German. The semantic structures were produced from Japanese titles of papers in information technology by the parser of Fujitsu's ATLAS/II-system, the German generator had to be designed and implemented by SEMSYN. A first prototype was presented at the end of 1984, further enhanced versions of this Japanese/ German system were demonstrated at various meetings, e.g. COLING-86 [Rösner 86a].

2. DESIGN GUIDELINES

When we designed and implemented the initial SEMSYN generator we tried to base this work on guidelines like the following [Rösner 86b]:

— The implemented system should not be confined by its first application. Since we had only little influence on the format of the output of our partner's parser for Japanese we designed a more general frame language that served as input to our system.

— The generator should be easily extensible and transportable to other applications.
This gave strong motivation for choosing an object-oriented implementation technique.

—The generator should be integrated in an environment of software tools supporting "every day" work (e.g. testing, debugging, experimentation) with the system.
This led to the implementation of a whole system of utilities: e.g. menu-based tools for lexicon update and maintenance or an interactive editor for semantic structures.

3. THE BASIC MACHINERY

The SEMSYN generator is organized into two major modules :
—the generator kernel or 'realization component' and
—the front end generator or 'morpho/syntactic component'.
We will have a closer look at the operation of these modules now.

3.1. The generator kernel

The generator kernel starts from a semantic representation, i.e. a 'message' in the sense of [McDonald et al. 87]. Its task is to 'realize' the message, i.e. to decide how its content may be expressed in natural language :
—What is the adequate syntactic form for the utterance as a whole?
— How should the subparts of the conceptual representation be realized and integrated into the utterance?
— What are appropriate lexicalizations – as lexemes or whole phrasal structures of the target language – for the elements of the message ?

3.2. The linguistic representation

The output of the generator kernel is a functional grammatical structure. This linguistic representation fully specifies the intended utterance :
—the syntactic category of the whole utterance and the grammatical functions and syntactic categories of all subparts.
—the syntactic features of the head of each syntactic entity,
— the lexemes or special lexical items – marked with category information like : *PN for proper names or: *NC for noun compounds – to be used.

3.3. The front end generator

The functional grammatical structures produced by the generator kernel are input to the front end generator. This module has to execute all syntactic and morphological processes that are necessary to produce the corresponding surface string. This involves:
—linearization, i.e. constituent ordering,
—agreement handling,
—inflection.

The need for an explicit linguistic representation of the intended utterance and a separate final processing step is especially obvious for highly inflectional languages with a rich repertoire of agreement phenomena (e.g. French, German).

3.4. Examples

3.4.1. *Frame structures as semantic representation*

SEMSYN's generator kernel expects its input in a frame notation. Although there are minor variations between the different applications the basic format is fixed: frame structures consisting of a 'semantic symbol' as name and named roles or slots with - recursively - frame structures as fillers. An example of a case frame:

```
(GENERATE
        : AGENT (PROJECT: NAME (: *PN SEMSYN))
        : OBJECT (LANGUAGE : ATTRIBUTES GERMAN))
```

Here the toplevel frame structure contains the semantic symbol 'GENERATE' and has two filled roles: AGENT and: OBJECT as further information.

3.4.2. *A realization result*

When the generator kernel realizes this case frame as a clause in active voice this results in the following functional grammatical structure :

```
(: CLAUSE
    (: VERB 'generier')
    (: FEATURES (: VOICE ACTIVE))
    (: SUBJ
        (: NG (: HEAD
            (: *NC (: *PN 'SEMSYN')
                    '.'   'Projekt')
        (: FEATURES
            (: NUM SG) (: DET DEF) (: CAS NOM))))
    (: DIROBJ
        (: NG (: HEAD 'Sprache')
            (: FEATURES (: NUM SG) (: DET ZERO))
            (: CLASSIFIER 'deutsch'))))
```

This yields the following German sentence:
"Das SEMSYN-Projekt generiert deutsche Sprache."
(The SEMSYN project generates German language.)

3.5. Object-oriented implementation of realization knowledge

The main features of the object-oriented paradigm that we exploited for the implementation of realization knowledge in the generator kernel are
— hierarchy as organisation principle for the knowledge base and
— message passing between objects as primary control structure.

The specialization hierarchy used is rooted in a general class that defines the basic methods for realization (KBS-Schema). On the next level there are general classes for
— case frames (CASE-Schema)
— concepts (CONCEPT-Schema)
— relations (RELATION-Schema).

These classes differ with respect to the possible realizations of their instances :
— concept-schemata allow only realizations as noun groups
— case-schemata allow for various clausal forms (active, passive, topicalized) as well as nominalized forms
— subclasses of relation-schema incorporate knowledge about realization possibilities for (more complex) semantic relations like the relation between: MEANS and: PURPOSE,: REASON and : RESULT etc.

3.5.1. *A relation-schema*

The semantic representation of a summary of Macbeth may contain the following frame structure:

```
(REASON-FOR
    : RESULT
        (MURDER: AGENT MACBETH : OBJECT DUNCAN)
    : REASON
        (PERSUADE
            : AGENT (LADY-MACBETH : SPECIALIZE AMBITIOUS)
            : OBJECT MACBETH))
```

One possible way to express this relation is to realize the fillers of: REASON and : RESULT as clauses and add the clause from : REASON as a subordinate to that of: RESULT:

"Macbeth ermordete Duncan, da die ehrgeizige Lady Macbeth Macbeth überredete".
(*Macbeth murdered Duncan because the ambitious Lady Macbeth persuaded Macbeth).*

4. OTHER APPLICATIONS OF THE SYSTEM

In the meantime improved and extended versions of the SEMSYN generation system have been applied to quite a variety of input structures and generation tasks:

— Machine translation applications :
- Generation of German from (handwritten) semantic structures proposed for use within EUROTRA [Heid, Rösner, Weck 87]
- Generation of German sentences in the domain of doctor/patient communication from semantic structures produced from Japanese and English by CMU's Universal Parser [Tomita, Carbonell 86].

— Text generation:
- SEMTEX: generation of news stories from statistical data [Rösner 87]
- GEOTEX: generation of descriptive texts for geometric constructions [Kehl 86].

Although the basic design of the generator [Rösner 86b] proved to be flexible enough and could remain untouched each of these applications has led to additional features of the whole system.

4.1. MT applications

4.1.1. *Title translation*

In the first application of the system we started from semantic representations derived from titles of Japanese papers in the field of information technology. Titles are in most cases noun groups. In order to generate German equivalents we had to provide the prototype primarily with knowledge about German noun group structures. On the other hand, for many of these semantic structures clausal forms were possible as well. We therefore provided the system with "stylistic" switches that allowed the alternative generation of clauses from case frames as well.

4.1.2. *MT for doctor/patient communication*

The sample of semantic structures in this experiment was taken from doctor/patient communication.

The semantic structures produced by CMU's parsers for Japanese and English are basically case frames, but include syntactic information as well (e.g. about: MOOD or: TIME). The fragment of German generable by the SEMSYN system was extended by yes/no-questions and imperatives.

An example:
English input to CMU's parser:
"i have a pain in the throat".

Semantic structure as input to SEMSYN :

```
("HNAVE-A-SYMPTOM
      : MOOD DEC
      : AGENT (* PATIENT
                  : HUMAN + : PRO 1: NUMBER SG
                  : PERSON 1)
       : TIME PRESENT
       : SYMPTOM (* PAIN
                       : LOCATION
                            * BODY-PART: NAME *THROAT)))
```

German generation:

"Ich habe Schmerzen im Rachen."

4.1.3. *EUROTRA-D/SEMSYN experiment*

In order to support the EUROTRA-D group, we ran this experiment:
A sample of semantic structures as proposed for use within EUROTRA [Steiner 86] should serve as input to our generator.

This experiment was interesting under various aspects :

— The semantic representation used is based on systemic grammar; since the classes used are already hierarchically structured it was relatively easy to implement them as a FLAVOR hierarchy of realization classes.

— The sample of semantic structures was chosen to cover the complete list of German sentential types from a textbook [Helbig, Buscha 86]. In order to be able to generate all of these surface forms we had to further enrich the generable fragments with e.g.

- infinitival complements
- genitive objects
- subject and object clauses.

4.2. Text generation

4.2.1. *SEMTEX: Generation of news stories*

SEMTEX starts from mere labor market data, extracts a list of semantic representations from them as "text plan" and then converts this list into texts like the following:

"Die Zahl der Arbeitslosen in der Bundesrepublik Deutschland ist im Dezember spürbar angestiegen. Sie hat von 2210700 auf 2347100 zugenommen. Die Arbeitslosenquote betrug Ende Dezember 9.4 Prozent. Sie hatte sich Ende Dezember des letzten Jahres auf 9.3 Prozent belaufen. Der DGB hat erklärt, er sehe in der Vergrößerung der Arbeitslosenzahl ein negatives Zeichen."

The main concern in implementing SEMTEX has been to provide the SEMSYN generator with mechanisms that keep track of previous generation decisions thus creating a representation of the textual context built up by the already uttered sentences. This context is used:
— to avoid repetition in wording,
— to deliberately elide information still valid (e.g. about the time period concerned),
— to decide on pronominalisation and other types of reference.

In addition a representation of the temporal context is used :
— to dynamically determine grammatical tense and
— to produce appropriate natural language descriptions for the time units mentioned [Rösner 86b].

4.2.2. *GEOTEX: Verbalizing objects and operations*

In the GEOTEX application the SEMTEX text generator is combined with a tool for interactively creating geometric constructions [Kehl 86]. The latter offers formal commands for manipulating (i.e. creating, naming and - deliberately - deleting) basic objects of Euclidean geometry. The generator is used to produce descriptive texts related to the geometric construction:
— descriptions of the geometric objects involved,
— descriptions of the sequence of steps done during a construction.

Verbalizing the course of a construction :

When GEOTEX is describing the course of a construction in a concise and coherent text it starts from the sequence of commands of the geometry language. Let us look at an example :

(PUM SA 15 10)
(PUM SB 20 7)
(KRE SK SB SA)

Each of these commands in turn causes GEOTEX:
— to update the associated FLAVOR representation for the domain,
— to display (if possible) the objects on the screen (in this case : point SA with coordinates (15, 10), point SB with coordinates (20, 7), circle SK with center SB and through SA),
— to create a message from the operation and give it as input to SEMTEX.

SEMTEX renders this information in the order given. For the example this resulted in the following text:

"Ich zeichne den Punkt Sa (15/10) ein."
(I draw point Sa (15/10.)
"Und den Punkt Sb (20/7)."
(And point Sb (20/7).
"Um ihn schlage ich den Kreis Sk durch Sa."
(Around it I draw Sk through Sa.)

To achieve this result SEMTEX' context-handling mechanisms have been enriched:
— Elision is no longer restricted to adjuncts. For repetitive operations verb and subject will be elided in subsequent sentences (cf. the sentences 1 and 2).
— The distinction between known information (i.e. known geometric objects) and new one (i.e. new objects created from known ones) is exploited to decide on constituent ordering : the constituent referring to the known object is "topicalized", i.e. put in front of the sentence (cf. sentence 3).

In addition the system allows for more ways to refer to objects introduced in the text: pronouns, textual deixis using demonstrative pronouns ("dieser Punkt", *this point),* names. The choice is done deliberately : pronouns are avoided if their use might create an ambiguity; reference by name is used when an object has not constantly been in focus and therefore has to be re-introduced.

5. SEMSYN'S SOFTWARE ENVIRONMENT

SEMSYN's generation system has been implemented on a SYMBOLICS lisp machine. During the implementation we aimed at utilizing as much of the functionality of this machine in order to get optimal support for our work. We have built up an environment of linguistic and software tools that, though designed for our projects purposes, may be - at least in part - of interest for other projects in MT and CL in general. (1)

5.1. Interface tools

This comprises all software that provides easy and comfortable communication with the system (even for casual users).

SEMSYN's user interface is centered around SEMNET-GRAPHICS, a tool for visualizing semantic nets - the starting point of the generation - as mouse-sensitive graphics [Rösner 86b]. The graphical representation is embedded in an interface "frame" [Weinreb, Moon 81] whose "panes" are displaying various intermediate structures - depending on the users chosen "frame configuration" - and the generation result.

5.2. **Experimentation tools**

These tools extend the capabilities of the user interface and are intended to enable and support experiments with the system.

SEMNET-EDIT is a tool for experimenting the generator by interactively editing semantic nets [Kehl 85]:

— modification of given semantic nets
— creation of semantic nets from scratch
— generation of German from created or modified semantic nets and/or their subnets.

Experimentation tools of this type are not only useful for purposes of debugging and system improvement but proved as well to be very helpful as comfortable means for introduction into the system's capabilities and limitations.

5.3. Lexicon tools

In every realistic application dictionaries play an important role as body of linguistic knowledge; the need for support in maintaining and updating them seems obvious.

SEMSYSTEM uses two types of dictionaries : a single German root form dictionary (with morpho/syntactic information) for the generator front end and so-called "realization dictionaries", that relate semantic symbols to German lexical items (root forms of verbs, nouns, adjectives,...) and that may vary for different applications of the generator. For both types of lexica there are window - and menu-based tools for maintenance.

6. PROSPECTS : FROM MONO- TO MULTILINGUAL GENERATION

6.1. Teaching English to the system

In a recent experiment (2) we changed and extended our generator system in such a way that - using the same representation for the different domains - the texts of SEMTEX and GEOTEX may be produced in English as well.

A system produced example text from the newspaper application :

Increase in the number of unemployed.
NÜRNBERG/BONN (cpa) DECEMBER 5,85. The number of unemployed in West Germany has increased slightly during November.
It has increased from 2148800 by 61900 to 2210700. At the end of November the unemployment rate had a value of 8.8 percent. At the end of the year-ago period it had a value of 8.7 percent. Gerd Muhr, the speaker of the DGB, declares, it sees a bad sign in the increase in the number of unemployed.

French will be the next target language ; we have started to prepare the morphological and syntactic data for such an experiment.

6.2. Related work

There is more recent work in "multilingual generation" from data :

— RAREAS, a system synthesizing weather forecasts from data provided by meteorologists (Kittredge et al. 86] is currently being equipped with French as second target language.

— Kukich's ANA, a system generating English stock market reports from Dow Jones data [Kukich 83], has a second tongue as well: the generation of French bulletins has been possible by replacing ANA's "linguistic module" with a French version - called FRANA [Contant 86] - while leaving ANA's other modules untouched (i.e. Fact Generator, Message Generator, Discourse Organizer).

6.3. Why multilingual generation ?

6.3.1. *Aspects of application*

Generation of natural language texts in different languages (and probably different styles) from the same knowledge base might be an interesting alternative to human or machine translation of these texts.

Re-Generation (of e.g. software manuals or maintenance handbooks) in different languages might be much more economic than manually "updating" those texts when the underlying knowledge base changes.

6.3.2. Aspects of implementation (3)

Multilingual generation enforces the separation of generator knowledge into language dependent data and language independent machinery.

In order to keep a generator easily portable to other languages the implementor will have to allow as much declarativity as possible.

6.3.3. *Aspects of linguistic theory*

Work in multilingual generation from semantic representations may be seen as an exercise in contrastive linguistics :

A central issue for any generator starting from semantic structures is the choice of an appropriate syntactic structure for the expression of a given meaning structure.

What are the differences and correspondencies between the different target languages with respect to this choice? (E.g. Focus may be expressed by constituent order in German, in English you may have to choose a passive.) *Similarly.*

What are the differences between the target languages with respect to the semantic features that are obligatory in order to be able to produce surface text? (E.g. In the Japanese/German MT application we were confronted with the fact that the semantic structures derived from Japanese did not contain semantic information about in/-definiteness or multiplicity.)

ACKNOWLEDGEMENTS

The SEMSYN project is funded by the West German Ministry for Research and Technologyh (BMFT). The project is currently cooperating with partners from Japan (University of Kyoto; NTT Basic Research Laboratories) and USA (International Center for MT at CMU). We have to thank all partners for their support

The SEMSYN system is the joint effort of a variety of people. Special thanks to M. Emele (Stuttgart) for his work on the front end generator, to W. Kehl (Stuttgart) for his implementation of GEOTEX and the editor for semantic nets and to O. Rambouw (Ithaca, N.Y.) for our joint experiment to teach English to the system.

REFERENCES

CONTANT C, "*Génération automatique de texte: application au sous-language boursier*", M.A. thesis, Dept. de Linguistique, Univ. de Montréal, 1986.

EMELE M., "*FREGE - Entwicklung und Implementierung eines objektorientierten FRont-End-GEnerators für das Deutsche"*, Diplomarbeit, Institut für Informatik, Uni Stuttgart, 1986.

HEID U; RÖSNER D.; WECK B.; 'Das EUROTRA-D/SEMSYN-Experiment: Generierung deutscher Sätze aus semantischen Repräsentationen', in: Tillmann, H.G.; Willée (Hrg.): "*Analyse und Synthese gesprochener Sprache"*, (Hildesheim: Olms), 1987.

HELBIG G.; BUSCHA J., *"Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht*", Leipzig, 1986.
KEHL W., "*Erweiterung der graphischen Schnittstelle des SEMSYN-Projekts",* Studienarbeit, Institut für Informatik, Univ. Stuttgart, 1985.
KEHLW., "*GEOTEX- Ein System zur Verbalisierung geometrischer Konstruktionen",* Diplomarbeit, Institut für Informatik, Univ. Stuttgart, 1986.
KEMPEN G. (Ed.), "*Natural language generation: New results in Artificial Intelligence, Psychology and Linguistics",* Kluwer Academic Publishers, Dordrecht/Boston, 1987.
POLGUERE R.A.; GOLDBERG E., "*Synthesizing Weather Forecasts from formatted Data",* in : COLING-86, Proceedings, Bonn, August 1986.
KUKICH K., "*Design and Implementation of a Knowledge-Based Report Generator",* ACL Annual Meeting, Proceedings, 1983.
McDONALD D.D., "*Natural Language Generation as a Computational Problem : an introduction",* in : Brady & Berwick (eds.) "Computational Models of Discourse", MIT Press, 1983.
MCDONALD D.D. ; PUSTEJOVSKY J.D. ; VAUGHAN M.M., "*Factors contributing to efficiency in natural language generation",* in : [Kempen 87].
RÖSNER D., "*When Manko talks to Siegfried - Experiences from a Japanese/German Machine Translation Project",* in : COLING-86, Proceedings, Bonn, August 1986a.
RÖSNER D., "*Ein System zur Generierung von deutschen Texten aus semantischen Repräsentationen", Dissertation, Institut f. Informatik, Univ.* Stuttgart, 1986b.
RÖSNER D., *"The automated news agency: the SEMTEX text generator for German ",* in: [Kempen 87]
STEINER E., "*Generating Semantic Structures in EUROTRA-D*", in: COLING-86, Proceedings, Bonn, August 1986.
TOMITA M. &CARBONELL J., "*Another Stride Towards Knowledge-Based Machine Translation",* in: COLING-86, Proceedings, Bonn, August 1986.
WEINREB D. & MOON D., "*LISP machine manual",* MIT, 1981.

NOTES

(1) These tools are best illustrated by an interactive demo.
(2) This work is done in collaboration with Odyssey Research Associates, Ithaca, N.Y.
(3) The SEMSYN generator and the applications as described in this paper are fully implemented and run in ZetaLISP and FLAVORS on SYMBOLICS lisp machines.

Address: Projekt SEMSYN, Institut für Informatik, Universität Stuttgart,
Herdweg 51, D-7000 Stuttgart l (West Germany).
Received: 12 December 1988.