A Framework for Lexical Selection in Natural Language Generation

Sergei Nirenburg
Carnegie-Mellon University

Irene Nirenburg
Carnegie Group Inc

Abstract. This paper describes a procedure for lexical selection of open-class lexical items in a natural language generation system. An optimum lexical selection module must be able to make realization decisions under varying contextual circumstances. First, it must be able to operate without the influence of context, based on meaning correspondences between elements of conceptual input and the lexical inventory of the target language. Second, it must be able to use contextual constraints, as supported by collocational information in the generation lexicon. Third, there must be an option of realizing input representations pronominally or through definite descriptions. Finally, there must also be an option of using elliptical constructions. The nature of background knowledge and the algorithm we suggest for this task are described. The lexical selection procedure is a part of a comprehensive generation system, DIOGENES.

1 Our place on the generation research map.

Natural language generation is traditionally divided into two stages: the utterance planning ('what to say') stage and the lexical and syntactic realization ('how to say it') stage. The latter stage consists, essentially, of a large set of realization choices for the various meanings of the input, using the morphological, syntactic and lexical means of expression in the target language (TL). Research reported here deals with the process of lexical selection during this second stage of generation. Many of the existing generation systems have been conceived as components of natural language interfaces to database systems. In such generators the lexical inventory can be strongly constrained without jeopardizing the quality of the interaction (cf., e.g., McKeown, 1985). Such systems necessarily concentrate on choosing appropriate TL syntax -- indeed, generators are expected to produce adequate syntactic structures. Lexical selection becomes more important when it is difficult to constrain the types of output in generation, and, consequently, when the lexicon becomes large. Machine translation and automatic text summarization are among applications that by nature require a wide range of outputs and have to use a sizeable lexicon. Note that of these two the former does not involve utterance planning and concentrates on lexical and syntactic realization.

In the natural language generation community the task of lexical selection has not yet attracted a sufficient amount of attention, though it was addressed in a well-known early generation project (Goldman, 1975) and is widely recognized as an important problem (cf. Danlos, 1984; Jacobs, 1985; Bienkowski, 1986; and the survey Cumming, 1986). One motivation for this research was that we agree with Marcus (1987, p. 211) that 'most generation systems don't use words at all,' and we believe that the quality of generation output will improve significantly once an adequate lexical selection component becomes a standard part of a NLG system.

2 The Task

Research reported in this paper was performed within the DIOGENES project (Nirenburg, 1987), whose objective is to provide a high-quality generator for a knowledge-based interlingual machine translation system. The *input* to this generator is a set of a) world concept instances that represent the propositional content of the original text, and b) sets of text parameter values that represent its pragmatic content. (These concepts are represented in a frame-oriented formalism and are interconnected according to the rules of a special grammar -- see Nirenburg et al., 1986 for a detailed description.) In this paper we deal with a subset of the generation task, namely, the selection of open-class lexical items to realize the meanings of object, event and property tokens in the input. Thus, the *output* of the generation module described here is a lexical unit or a pronoun in the target language.

Our approach (and especially the expected input) to text generation is similar to that of the SEMSYN project (e.g. Rösner, 1986). Lexical selection is not, however, an immediate concern of and is not discussed at any length in SEMSYN descriptions (see, for instance, Laubsch et al. (1984, p. 492), and a published analysis of practical difficulties encountered by the project (Hanakata et al., 1986) does not address this issue at all. Furthermore, since until very recently that project had to generate sentence-length texts (article titles), the problem of definite descriptions, pronominalization and ellipsis did not become acutely important.

3 Why is it a difficult task?

Lexical choice is not a straightforward task. Suppose we have to express in English the meaning 'a person whose sex is male and whose age is between 13 and 15 years.' What knowledge do people use in order to come up with an appropriate choice out of such candidate realizations as those listed in (1).

(1) *boy, kid, teenager, youth, child, young man, schoolboy, adolescent, man,*

Without a sentential context the choice, based on closeness of the meaning match and generality of meaning, should be *boy*. For a computer program to be capable of making choices like this, it has to possess a preference-assigning capability on the matches between the meanings of the candidate lexical realization on the one hand and the input meaning unit (see the discussion of the matching metric below).

3.1 Collocations

Lexical choices are, however, typically made in context. Contextual relations among lexical units reflect meaning-induced constraints on cooccurrence (selectional restrictions: *admire* takes a human subject). Sometimes, however, it is difficult to formulate a cooccurrence constraint in terms of selectional restrictions alone. Thus, for example, the causative construction with the English *influence* requires *exert*; its Russian equivalent *vlijanie* requires *okazyvat'*, and the latter is not a Russian correlate of *exert* other than in the above and very few similar syntagmatic constructions. Why do we use, in English, *shed* with *tears* or *leaves* but don't usually say *shed water out of a bucket* or *they drop tears every time when* <...>? Such properties of the lexical stock of a natural language are called *collocational*. We will now illustrate the concept of collocation through several examples.

Consider the conceptual operator of *a large quantity of*, a (relative) value for measuring quantities (of materials, forces, qualities, properties, etc.). It is realized in English in accordance with collocational properties of the lexical units that are used as its operands. Not every quantity goes with every realization of the above operator. Members of the set <*big, enormous, great, high, large, strong, wide*> of potential realizations of *a large quantity of* can cooccur with every of the members of the set <*amount, difficulty, expanse, selection, voltage*> of quantities. We say *high voltage* but *a large amount*. It would be inappropriate for a generation system to produce something like *high selection* or *large difficulty*. (Note that in *parsing* the problem of assigning a similar semantic marker to all the various expressions from the example can, in principle, be tackled through a mechanism of metaphor processing (e.g., Carbonell, 1987), whereby a *general heuristic rule* is developed for processing metaphorical input belonging a single class, such as, for instance, *a large quantity of...* -- see Lakoff and Johnson, 1980, for an extensive listing of potential metaphor classes; in generation, however, the task is the opposite -- to *produce* fluent metaphorical language. Since this depends not on regularities of meaning, but rather on the idiosyncrasies of meaning realization in the various natural languages, the general rules will be more difficult to come by and formulate.)

An additional class of collocations are the paradigmatic collocations. These are best exemplified by the 'set-complement' collocations such as the English *left* and *right* or *parents* and *children*. The knowledge of these collocations, for instance, simplifies the process of lexical selection of conjoined constructions, such as *ladies and gentlemen*.

Collocational relations are defined on lexical units, not meaning representations. The study of collocations ascends to Firth (1951); it is

a central part of the *Meaning - Text* school of linguistics — cf. Mel'čuk, 1974; 1981. The importance of collocational properties in generations has been recognized (cf. Cumming, 1986), but relatively few systems actually include collocational information in their decision processes.

3.2 Ellipsis and Anaphora

Certain contexts completely alleviate the problem of open-class lexical selection. Consider the following (gloss of an) input segment:

(2) $Clause_1$: Buy($John_3$ $book_7$), $time_1$, focus: $book_7$
$Clause_2$: Bring($John_3$ $book_7$ $office_1$), belong-to($office_1$ $John_3$), $time_2$: $time_2 > time_1$, focus: $office_1$
$Clause_3$: Read($John_3$ $book_7$), aspect: inchoative, time: after($time_2$)

One of the adequate ways of realizing it is:

(3) *John bought a book. He brought this book to his office and started to read it.*

There are seven instances of the three object-type concepts in the case-role slots of the input propositions above. Each of the three concepts is realized lexically only once. In two cases these meanings were realized through pronominalization and in one each through definite description and an elliptical construction. This example shows that non-lexical realization is an integral part of the process of lexical selection in generation.

In what follows we briefly describe the system architecture, the knowledge structures and the algorithm we use for selecting open-class lexical items in generation.

4 The System and the Knowledge

DIOGENES is a distributed natural language generation system featuring a blackboard-type control structure. The processing in it is concentrated in the *knowledge sources* which are triggered by the state of the various blackboards. The latter contain the *input* to generation as well as all intermediate and final results of DIOGENES operation, represented uniformly in a frame-oriented knowledge representation language. Background knowledge in DIOGENES includes the following components relevant to the task of lexical selection:

- a *concept lexicon*, a set of knowledge structures that describe object and event-types in the (sub)world of the texts to be generated (the first application of DIOGENES is, for example, in the domain of computer hardware manuals)
- a *generation lexicon* that links (sub)world concepts (or, more accurately, their instances) with particular lexical units of the target language.

The above description is necessarily incomplete. See Nirenburg, 1987 for an extensive specification of all the facets of DIOGENES.

The implementation vehicles for DIOGENES are the Framekit knowledge representation language (Carbonell and Joseph, 1985) and CMU CommonLisp running on an IBM PC RT.

Sample concept lexicon entries are illustrated in Figure 1. The figure shows a screen of the knowledge acquisition and maintenance system, called ONTOS (Nirenburg et al. 1988), which we use for acquiring and maintaining the lexicons. The figure shows a partial view of the concept network and three concept lexicon frames corresponding to the concepts of *research-workstation, memory* and *disk*.

The following is a sample input that will allow DIOGENES to produce the sentence

The basic IBM personal computer XT consists of a system unit and a keyboard

```
((ID clause1)
 (PROPOSITION part-of1)
 (MODALITY real)
 (SUBWORLD computer-world)
 ((SPEECH-ACT definition)
  (DIRECT? no)
  (SPEAKER author)
  (HEARER reader))
 (FOCUS
   (GIVEN role1)
   (NEW (and role2 role3))))
((PROPOSITION part-of1)
 (IS-TOKEN-OF part-of)
 (PATIENT role1)
 (COUNTERPATIENT ((and role2 role3))
 (ASPECT
  (PHASE begin-end)
  (DURATION always)))
```

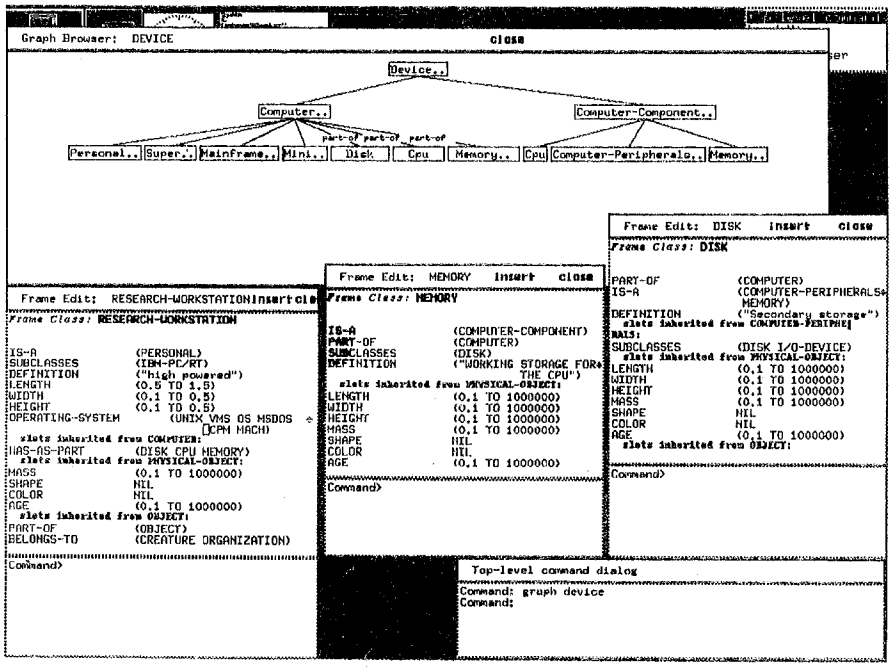

Figure 1. Concept Lexicon Entries

```
((ID role1)
 (EXTENSION IBM-PC1)
;an object instance to which all the various
;descriptions (intensions) of it refer;
;for example, "John Smith" can be intensionally
;represented as "John," "Mr. Smith," or "Jim's
; father" ;-- but it will refer to the same
;extension
 (INTENSION
  (IS-TOKEN-OF IBM-PC)
  (QUANTIFIER universal)
  (SUBWORLD computer-world)
  (MODEL XT)
  (CONFIGURATION minimal)))
;"basic" means "minimal" set of components
;that can be called a PC; the best way of
;treating this is to define, in the ontology, an
;attribute "configuration" whose domain will be
;(car house computer ...) -- anything that has a
;basic price and extras, and whose range will be,
;for the time being, (minimal regular extra)


((ID role2)
 (EXTENSION computer-system-unit1)
 (INTENSION
  (IS-TOKEN-OF computer-system-unit)
  (QUANTIFIER universal)
  (PART-OF IBM-PC model = XT)
;one needs this tautology; otherwise, system
;units have to be concept NAMES in the
;ontology, note the binding for the "model"
;which does the same compositionally, without
;proliferating names
  (SUBWORLD computer-world)))

((ID role3)
 (EXTENSION computer-keyboard1)
 (INTENSION
  (IS-TOKEN-OF computer-keyboard)
  (QUANTIFIER universal)
  (PART-OF IBM-PC model = XT)
  (SUBWORLD computer-world)))
```

4.1 The Generation Lexicon

The main static knowledge source for generating of open-class items is a specialized generation lexicon (GL). The structure of an entry in the generation lexicon in DIOGENES is shown in Figure 2 (the BNF is incomplete wherever obvious):

The *importance value* serves to distinguish the saliency of the various relations for the identity of the entry head. Thus, for instance, generating *youth* instead of *boy* seems to be less a deviation than generating *girl*. This is why the importance of the *sex* slot in the example below is greater than that of the *age* slot.

The sample GL entries below do not contain a full complement of collocation relations.

```
(make-frame toss
  (is-token-of (value throw))
  (direction (value up)
     (importance 3))
  (altitude (value high)
    (importance 3))
  (velocity (value low)
    (importance 9))
  (object (value coin)
  (lexeme (value "toss"))
  (syntactic-info
    (lexical-class verb)
    (verb-type transitive)
    (morph regular)
  (para-collocation
    (antonym catch)
    (synonym cast propel toss
             fling hurl pitch pass)))

(make-frame new
  (is-token-of (value age.CL))
  (age (percent-of-range (0 25)))
  (domain (value non-living.CL))
  (lexeme (value "new"))
  (syntactic-info (lexical-class adjective))
  (morphological-info (comparative regular)
                      (superlative regular))
  (para-collocation (antonym old)))

(make-frame boy
  (is-token-of (value person.CL))
  (sex (value male)
       (importance 10))
  (age (value (2 15))
       (importance 4))
  (lexeme (value "boy"))
  (para-collocation (synonym lad kid child)
    (antonym girl adult)
    (hypernym person))
  (syn-collocations-in (value boy.syn)))

(make-frame boy.syn
  (agent-of (value play throw run jump)
    (strengh 0))
  (place (value school playground ballfield)
 (strength 0)))
```

```
GL-entry            ::= ( <meaning-pattern><TL-pattern>* )
<meaning-pattern>   ::= ( (token-of (value <CL-concept>))
                          [( <relation> (value <value>*)
                            (importance <importance-value>))]* )
<CL-concept>        ::= {any concept in concept lexicon}
<relation>          ::= {any relation from Concept Lexicon}
<value>             ::= {any concept or attribute (scale)
                         value in Concept Lexicon}
<importance-value>::= 1 | 2 | ... | 10
<TL-pattern>        ::= (<TL-lexeme><lex-info><collocation> )
<TL-lexeme>         ::= (<language>TL-lexical-unit | (synonym TL-lexical-unit*))
<language>          ::= english | spanish | japanese | ...
<lex-info>          ::= ((<syntactic-info>)(morph <inflection-type>))
<syntactic-info>    ::= {the contents of a syntactic dictionary
                         (cf. e.g. Ingria, 1987)}
<inflection-type>   ::= {an indication of irregularities in forming word forms,
                         e.g., @i[goose] - pl. @i[geese]}
<collocation>       ::= ( {<dimension> <dimension-value>*}* )
<dimension>         ::= {the name of a (syntagmatic or paradigmatic)
                         collocation relation based on the CL slot names
                         for the concept in question}
<dimension-value>   ::= {a TL lexical unit (word or expression)
                         that can ordinarily collocate with the
                         TL lexical unit in <TL-lexeme> above
                         and connected to the TL unit on a
                         specified dimension; can be recursive}
```

Figure 2. The Structure of the Generation Lexicon.

5 The Algorithm

In the DIOGENES generator an instantiation of a head-selecting knowledge source is triggered simultaneously for every *event* and *role* instance in the input representation. The results of their operation are posted to a public blackboard, so that all knowledge source instances can draw on this knowledge in their own decision processes. The knowledge sources responsible for selecting modifiers are triggered when the heads of their phrases have already been selected.

Figure 3 illustrates the algorithm for a single lexical selection (head or modifier) knowledge source. If an input frame was already mentioned in the input, the question arises whether it should be realized non-lexically, that is, using deictic means (this is the case with the second appearance of *John* in (2)). If so, a proper realization must be found and posted on the corresponding blackboard. If this process fails at any point, we revert to the 'regular' case of lexical realization. This latter consists, first of all, in scanning the generation lexicon in search of a set of candidate realizations for the input frame. (1) is an example of such a set. When such a set is produced, we attempt to filter it by removing those candidates that are not compatible with realizations already decided upon for other input frames in the same sentence. This processing is based on comparing the collocation information in the lexicon entries for the members of various candidate realization sets. For example, if a neighbor frame has already been realized as *demonstrator*, then the collocational information will filter out all members of (1) but *youth, teenager, man*. If the residual set has cardinality one, we post the result. Otherwise — as in the case when no collocational information can be used — we proceed to select the realization based solely on the entries in the candidate realization set (that is, without the

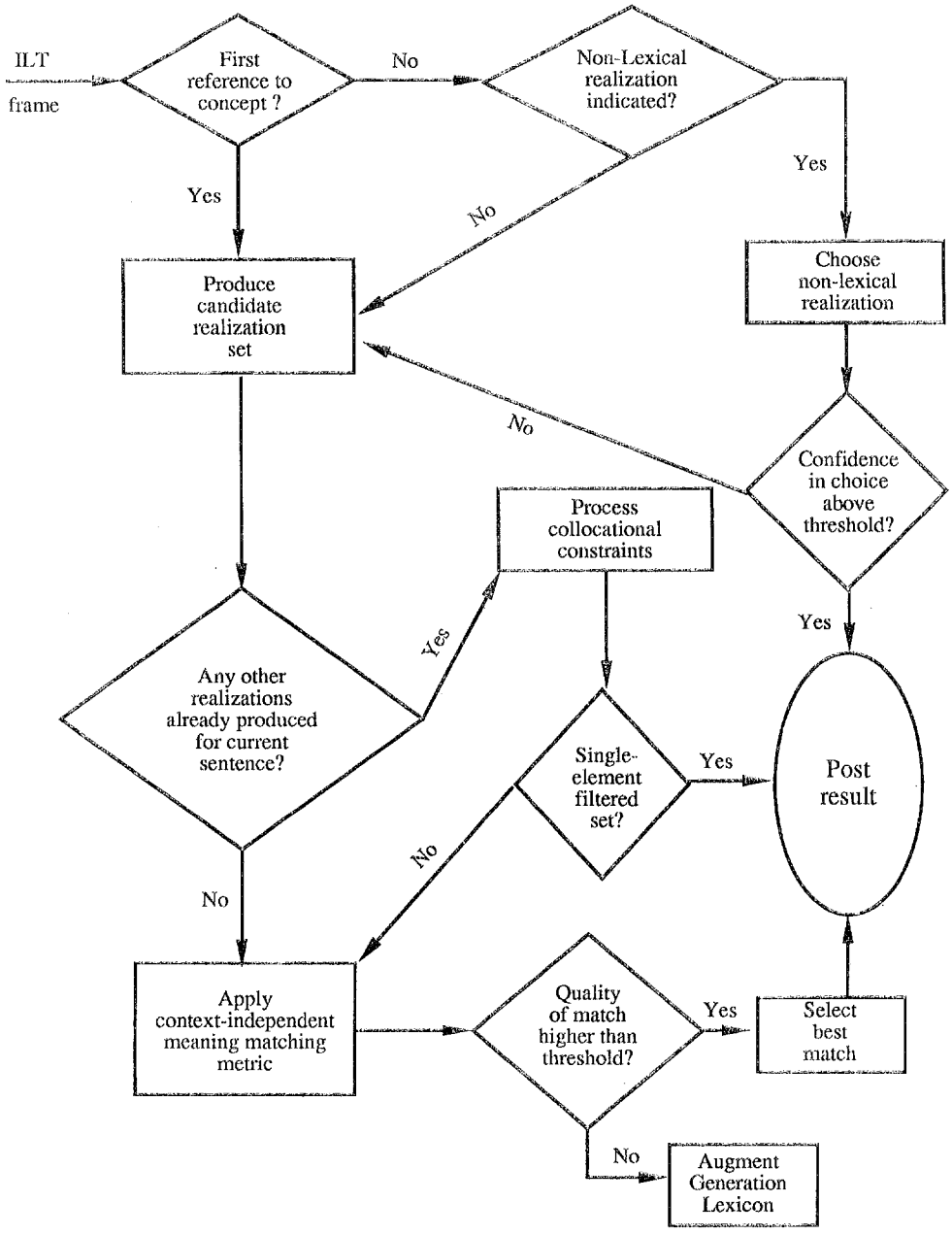

Figure 3. A procedure for selecting open-class lexical items during text generation. Incorporates the capability to introduce anaphora and ellipsis. Takes into account collocational knowledge for producing contextually appropriate realizations.

benefit of a context). This routine uses a well-defined inexact matching metric that calculates distances between the meaning of the input frame and the meanings of the lexical units in the candidate realization set. The closest meaning is then selected and posted.

6 Status and Future Work

The blackboard architecture and the inexact meaning matching module has been implemented; the collocation treatment module has also been implemented, but extensive testing has not been performed due to the lack of a large-scale lexicon. The anaphora treatment module has been implemented for pronominalization only, and the number of pronominalization rules employed has to be and will be increased.

It is clear that the acquisition of the generation lexicon is a major and extremely labor-intensive task in natural language generation. The acquisition of this dictionary, especially of the collocational information cannot at present be done automatically. But the efficiency of the team of human lexicographers working on this problem can be increased dramatically through the use of specialized intelligent interactive aids. We have developed one such *Knowledge Base Maintenance System* (cf. Nirenburg et al., 1987) for the acquisition of concept lexicons and will extend it so that it becomes applicable to the task of acquiring generation lexicons as well.

Acknowledgements

The authors would like to thank Victor Raskin, James Pustejovsky, Rita McCardell, Carl Pollard, Eric Nyberg, Scott Huffmann, and Ed Kenschaft for fruitful discussions of the topic.

References

Bienkowski, M.A. 1986. A Computational Model for Extemporaneous Elaborations. CSL Report 1, Cognitive Science Laboratory, Princeton University.

Carbonell, J. and R. Joseph. 1985. FRAMEKIT, A Reference Manual. Department of Computer Science. Carnegie-Mellon University.

Cumming, S. 1986. The Lexicon in Text Generation. Paper presented at the LSA Linguistics Institute Workshop on Lexicon, New York.

Danlos, L. 1984. Conceptual and Linguistic Decisions in Generation. In: Proceedings of COLING-84, pp. 501-504.

Firth, J.R. 1951. Modes of Meaning. In: J.R. Firth, **Papers in Linguistics**, London.

Goldman, N. 1975. Conceptual Generation. In: R. Schank (ed.), **Conceptual Information Processing**. Amsterdam: North Holland, pp. 289-371.

Hanakata, K., A. Lesniewski, S. Yokoyama. 1986. Semantic-Based Generation of Japanese-German Translation System. Proceedings of COLING-86. Bonn, pp. 560-562.

Hovy, E. "Integrating Text Planning and Production in Generation," in Proceedings of IJCAI-85, Los Angeles, 1985.

Ingria, R. 1987. Lexical Information for Parsing Systems: Points of Convergence and Divergence. In: D. Walker, A. Zampolli and N. Carzolari (eds.) **Automating the Lexicon: Research and Practice in a Multilingual Environment.** (in print).

Jacobs, P. 1985. A knowledge-based approach to language production. Ph.D. dissertation, University of California at Berkeley.

Laubsch, J., D. Rösner, K. Hanakata, and A. Lesniewski. 1984. Language Generation from Conceptual Structure: Synthesis of German in a Japanese/German MT Project. Proceedings of COLING-84. Stanford. pp. 491-494.

McKeown, K. 1985. **Text Generation**. Cambridge University Press.

Mel'čuk, I.A. 1974. **Towards a Theory of Linguistic Models of the Meaning-Text Type**. Moscow: Nauka.

Mel'čuk, I.A. 1981. Meaning – Text Models: A Recent Trend in Soviet Linguistics. **Annual Review of Anthropology**. Vol. 10, pp. 27–62.

Nirenburg, S., V. Raskin and A. Tucker. 1986. On Knowledge-Based Machine Translation. Proceedings of COLING-86. Bonn, pp. 627-632.

Nirenburg, S. 1987. A Distributed System for Language Generation. Technical Report CMU-CMT-86-102. Carnegie-Mellon University. May.

Nirenburg, S., I. Monarch, M. Calvin and T. Kaufmann. 1988. ONTOS: A Knowledge Acquisition and Maintenance System. CMU-CMT Internal Memo.

Rösner, D. 1986. When Mariko Talks to Siegfried. Proceedings of COLING-86. Bonn, pp. 652-654.