[From: Internat.conf. *Methodology & Techniques of Machine Translation*, British Computer Society, Cranfield, 13-15 February 1984]

THE GENERATION OF CHINESE SENTENCES FROM SEMANTIC REPRESENTATIONS OF ENGLISH SENTENCES

Xiuming Huang
Institute of Linguistics
Chinese Academy of Social Sciences
Beijing, China*

ABSTRACT

The paper describes the CASSEX package, a parser which takes as input English sentences and produces semantic representations of them, and gives an account of the generation procedure which translates these semantic representations into Chinese sentences.

0. Introduction

A Natural Language (NL) generator can be a system on its own right, as is (Meehan 76)'s TALE_SPIN which generates stories. More usually, however, a generator is part of a larger system, which generates surface text from an intermediate data structure produced by another component *of* the system, the analyser.

The generation component of a NL system plays a twofold role: firstly, it tests whether or not the output of the analysis component is correct, thus providing a kind of feedback to the analyser writer. For instance, (Goldman 75)'s generator BABEL detects that in the PARAPHRASE MODE, (Schank 75)'s conceptual analyser MARGIE fails to find the "reader" of the book:

INPUT: Reading the book reminded Rita to loan the book to Bill.
OUTPUT: Rita remembered to give the book to Bill and she expects him to return it to her because someone read the book.

Secondly, if the analysis output is correct, it tests whether or not the representation is good, in terms of the cost and efficiency involved in getting the final result usable to the user (inferences, paraphrases, summaries, answers, or translations, depending on the purpose of the system). Therefore, although generation has "traditionally been the poor relation in NL work" (Cater 81, p.30), a good generator is obviously a necessity to all NL workers.

For generating surface text from an intermediate data structure, we can either employ a connected body of grammar rules, most often an ATN generation grammar (Goldman 75, Simmons and Slocum 72, and Burton 76), sometimes an ATN for both analysing and generation (Shapiro 82); or we can use a set of functions or specialists (Boguraev 79, Cater 81).

* Mailing address: Cognitive Studies Centre, University of Essex,
Colchester C04 3SQ, England.

The generation procedure described in this paper takes the latter approach of using a set of functions because it is more straightforward and more economical to implement (you don't need another interpreter to run the generation ATN, for instance). Used in conjunction with the CASSEX package, an English sentence analyser, the generator produces good quality Chinese translations for a group of English sentences all of which contain the conjunction "and". The analyser and the generator comprise a prototype English-Chinese Machine Translation (MT) system. In this paper I will review the CASSEX package first, then give a description of the generation procedure.

1. The CASSEX package

1.1 An Outline of Boguraev's System

The CASSEX package is a parser developed from (Boguraev 79)'s work, a system based on ATN grammars (Woods 73) and Preference Semantics (Wilks 75). Boguraev's major aim was to resolve linguistic ambiguities, either lexical or structural in individual sentences. The resolution of ambiguities is shown by generating paraphrases for input sentences. Referential ambiguities, as well as ambiguities caused by conjunctions, were not taken into consideration in his system.

The overall design of Boguraev's system bears a strong resemblance to that of Winograd (1972). "The analyser ... seeks to use strong semantic judgment within the framework supplied by syntactically-driven parsing" (Boguraev 79, p. 0.2). Semantics routines (NPBUILD and SBUILD) are called after the system's syntax parser (an ATN) recognizes a noun phrase, relative clause, complement or *a* complete sentence. They fulfill two tasks:

1. Structurally, constructing for every input sentence one or more semantic representation(s) which is a dependency tree with verb as the most important node and case slots as its daughters (discussed in more detail in Section 1.4).

2. Judgementally, ruling out ill-formed semantic structures which blocks syntactically valid paths. In other words, the semantic routines confirm or block the syntactic paths of the parser - they never drive the parser (i.e. never suggest a particular syntactic path).

In the following sub-sections we will look at some of the system's features.

1.2 The Resolution of Word-sense Ambiguities

Word-sense ambiguities are resolved by the semantic routines. For instance, the sentence

(1) The green crook kicked the ball

could have sixteen possible interpretations, if we assume four word-senses for "green" ("green-coloured", "inexperienced", "angry", and "unripe"), two for "crook" ("shepherd stick" and "villain" and two for "ball" ("a spherical object for playing with" and "a social event for dancing"); however, NPBUILD delivers only two interpretations of "the green crook" for later processing, the two corresponding to "the inexperienced villain" and "the green-coloured shepherd stick"; then, after two readings of "the ball" are built, SBUILD is called and only one interpretation for the sentence is constructed, which is valid both syntactically and semantically, and reads to the effect that "The inexperienced villain kicked the spherical object".

1.3 The Treatment of Prepositions

Apart from tackling the problem of attaching prepositional phrases to appropriate constituents (mainly noun phrases or verbs), Boguraev attempts to handle all the ways in which prepositions can occur in a sentence. These are as:

- the particles of particled verbs, such as "away" in "throw away";
- in semi-idiomatic expressions like "green with envy" or where with a particular verb different prepositions impose different meanings on the verb, or express a finer distinction of meaning, e.g. "aim at" vs "aim for";
- in obligatory cases, e.g. "look at", "look for", etc.
- in optional cases, e.g. "go to the theatre with somebody", "rise with the sun", etc.

Boguraev's dictionary design allows for the first three types (these are not yet implemented in the CASSEX package); the fourth is handled by preplates, an adaptation of (Wilks 75)'s paraplates.

The preplates allow not only for the modification of verbs(as paraplates did) but also nouns. The structure of preplates is the same for both verbs and nouns. The following is the preplates for "with"*:

```
((*ENT ATTRIBUTE *INAN)  WITH1)
((MOVE   INSTRUMENT  THING)   WITH2)
((NOTHAVE MANNER *MAR)  WITH3)
((STRIK  INSTRUMENT *INST)  WITH4)
((CHANGE   INSTRUMENT *INST)  WITH5)
((CAUSE   INSTRUMENT *INST)  WITH6)
(((SEE   SENSE)   INSTRUMENT  (SEE THING))  WITH7)
((*DO MANNER *MAN)  WITH8)
((*HUM ACCOMPANIMENT *HUM)  WITH9)
((*DO ACCOMPANIMENT *HUM)  WITH10)
```

The actual preplate contains three elements. The first is the

* "WITH1", "WITH2", etc, are attached to the original preplates in Boguraev's system so as to meet the need of generating Chinese in later stage. "With" appearing in different preplates can have different equivalent in Chinese. In the following text, when talking about preplates, we will mean the actual preplate triples.

preferred semantic category of whatever constituent is being modified (a verb or a noun phrase); the second is the case relation between verb (or NP) and the postmodifying PP; the third is the required semantic category of the head noun of the postmodifying PP.

To show how preplates works in attaching PPs, consider the sentence

(2) I hit the man with the hammer.

The PP, or rather the head of the PP, is "hammer". Its head primitive is INST. The PP can either modify the verb "hit", whose head primitive is STRIK, or the NP "the man", whose head primitive is MAN.

Two of the above preplates match. Firstly, the preplate (*ENT ATTRIBUTE *INAN) because MAN ("the man") is an *ENT and INST ("the hammer") is INAN. Hence, the PP is tied to the NP: "the hammer" is an ATTRIBUTE of "the man". The sentence could be paraphrased as "I hit the man who had the hammer". Secondly, (STRIK INSTRUMENT INST) because the head primitives of "hit" and "hammer" are STRIK and INST. The PP is tied to the verb; "the hammer" has an INSTRUMENT relation to the verb. According to this case relation and PP attachment, the sentence could be paraphrased as "I hit the man with the hammer that I had".

1.4 The Semantic Representations Delivered by the Parser

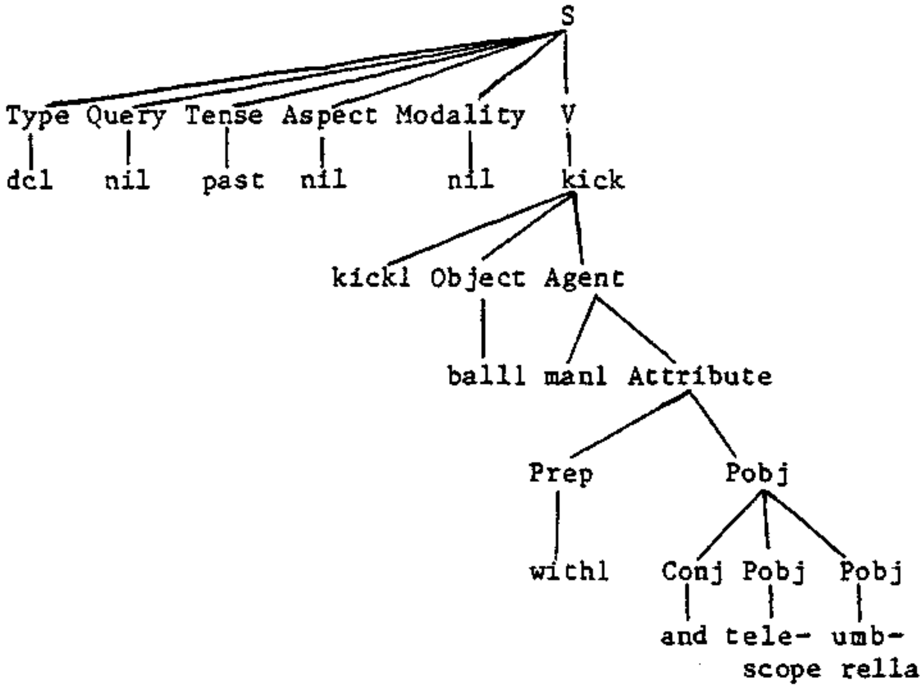
As was mentioned earlier, the semantic representations delivered by the CASSEX package are dependency trees with verbs as the most important nodes and case slots as their daughters. The representation for sentence (1) "The green crook kicked the ball" is as follows:

```
(CLAUSE (TYPE  NIL)
        (QUERY NIL)
        (TNS  PAST)
        (ASPECT  NIL)
        (MODALITY NIL)
        (NEG  NIL)
        (V (KICK ((*ANI SUBJ) ((*PHYSOB OBJE) ((THIS (MAN PART)) INST)
                  STRIK))
                 (OBJECT ((BALL1   (NOTFLOW THING))
                           (NUMBER SINGLE)
                           (QUANTIFIER SG)
                           (DETERMINER ((DET1   ONE)))))
                 (AGENT ((CROOK1(((NOTGOOD ACT)OBJE)DO) (SUBJ MAN))
                          (STATE (GREEN4 ((MAN POSS)(((NOTMUCH (TRUE
                                           THINK))   (SUBJ KIND))))))
                          (NUMBER  SINGLE)
                          (QUANTIFIER SG)
                          (DETERMINER ((DET1   ONE))))))))
```

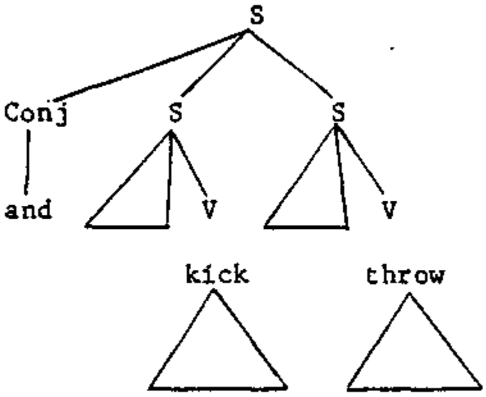
These representations, as we can see above, clearly show the syntactic structure and case relations between word-senses within constituents and between constituents. The surface sentence, together with the word order, however, has been lost: we don't to carry it along, like many MT system do (e.g. Liu 81), because dependency trees provide enough information for generating Chinese.

1.5 Processing Conjunctions

The major improvement of the CASSEX package over Boguraev's system is its ability to process conjunctions. In order to achieve this, grammars specifically designed for conjunctions have been incorporated into the system (see Huang 83 for detail). The CASSEX package deals with sentences containing Gapping, Right Node Raising or Reduced Conjunction, as well as the common cases of "and" conjunction. As for the representation of conjunctions, I follow (Ross 67)'s line, treating them as sisters of the conjuncts. The following are two examples.



(3) The man with the telescope and the umbrella kicked the ball.



(4) The man kicked the ball and the woman threw the ball.

2. The Generator

2.1 Boguraev's Generator

The generation procedure in Boguraev's system is used for providing paraphrases of the original input sentences. It contains three main steps:

1. Selection of the main verb from a set of verbs synonymous with the verb-sense in the semantic representation given by the analyser, and selection of the rest of the target language words (here the target language is English). This step reduces the number of possible output verb synonyms to just one.

2. Definition of the syntactico-semantic relationships. This is realised by the production of an environment network which contains both syntactic and semantic information relevant to the contextual environment (i.e., the information stored in the Wilksian word-sense formula) of the main verb.

3. Actual output of the generated sentence. This phase makes extensive use of the target language dictionary and grammar rules and makes sure that the generated sentence is a syntactically well-formed string of words.

The generator works impressively, producing well-formed paraphrases for many ambiguous sentences.

2.2 The Chinese Generator

Boguraev's generator doesn't suit our purpose very well, however, for several reasons. First, it was written for paraphrasing in English, hence its verb-centred nature (emphasis on main verb selection; the production of the environment network around the verb). In Chinese, the verb is less important (you can have sentences without verbs at all), while word order plays a vital role. Second, it is unable to handle coordinate constructions. Last but not least, it could have been written in a more concise and more straightforward way (at least for the purpose of generating Chinese).

Our generator is composed of a set of LISP functions listed below:

```
GENERATE
  GEN_SENTENCE
    GEN_CLAUSE
    GEN_STN_HEAD
    GEN_SUBJECT
    GEN_VERB
    GEN_OBJECT
    GEN_INDOBJ
    GEN_DOBJ
    GEN_MOBJ*
    GEN_POST_VERB_MOD
```

```
          GEN_RESULT_MOD
          GEN_GOAL_MOD
   GEN_STN_TAIL
```

The top one, GENERATE, takes as its argument a semantic representation and returns as output a Chinese sentence. It sets a global variable STN_SUBJ for later use (conjunction reduction), and calls a function STN_TAIL to get the appropriate sentence ending punctuation. The main function GEN_SENTENCE is called within GENERATE. It checks whether there is a conjunction at clause level; if there is, it calls GEN_SENTENCE recursively to process the conjuncts one by one (each conjunct may itself be comprised of a conjunction and two or more clause-conjuncts). Then we have the basic clause constructor, GEN__CLAUSE, which outputs single clauses. We decompose GEN_CLAUSE into specialists for constructing the major constituents of the clause: GEN_SUBJ, GEN_VERB, GEN_OBJ and GEN_POST_VERB_MOD. The building blocks needed for those specialists (i.e., noun phrases, preposition phrases, adjective phrases, etc.) are supplied by functions GEN_NP, GEN_PP, and GEN_ADJP.

2.3 The Clause Constructing Function

The Chinese language is basically an SVO language, though there are cases where the pattern SOV or OSV or even OVS occurs. We can rewrite any Chinese sentence in an SVO pattern while maintaining the fundamental meaning structure of the sentence. A text containing such sentences may be boring to read, but the economy achieved within CASSEX by having only one sentence pattern is much more important to us. Therefore, in our generation procedure, a uniform pattern SVO is assumed. This determines the definition of the function GEN_CLAUSE:

```
(DE GEN_CLAUSE
     (LET (AGENT_STR (ASSOC 'AGENT V_STR)
          (OBJECT_STR (ASSOC  'OBJECT V_STR)
          (MOBJECT_STR (ASSOC  'MOBJECT V_STR)
          (RECIPIENT_STR (ASSOC  'RECIPIENT V_STR)
          (RESULT_STR (ASSOC   'RESULT V_STR)
          (GOAL_STR (ASSOC 'GOAL V_STR))
     (APPEND   (GEN_STN_HEAD)   (GEN_SUBJ)   (GEN_VERB)   (GEN_OBJECT)
               (GEN_POST_VERB_MOD)]
```

A few explanations are needed to make the function more comprehensible. Suppose we are working on the semantic representation for (1) in Section 1.4; V_STR will be

```
(KICK ((*ANI  SUBJ) ((*PHYSOB OBJE) ((THIS (MAN PART)) INST) STRIK))
      (OBJECT  ((BALL1  (NOTFLOW THING))
               (NUMBER  SINGLE)
               (QUANTIFIER SG)
               (DETERMINER ((DET1   ONE)))))
      (AGENT  ((CROOK1 (((NOTGOOD  ACT) OBJE) DO) (SUBJ MAN))
```

* As the "that" clause in "John admitted to Bill that he loves Mary".

```
                (STATE (GREEN4 ((MAN POSS) (((NOTMUCH (TRUE
                                 THINK)) (SUBJ KIND))))))
                (NUMBER  SINGLE)
                (QUANTIFIER SG)
                (DETERMINER ((DET1 ONE))))))
```

AGENT_STR will be

```
        (AGENT ((CROOK1   (((NOTGOOD ACT)   OBJE)   DO)   (SUBJ MAN))
                (STATE (GREEN4  ((MAN POSS)   (((NOTMUCH  (TRUE
                                 THINK))   (SUBJ KIND))))))
                (NUMBER  SINGLE)
                (QUANTIFIER SG)
                (DETERMINER ((DET1   ONE)))))
```

OBJECT_STR will be

```
(OBJECT  ((BALL1 (NOTFLOW THING))
               (NUMBER  SINGLE)
               (QUANTIFIER SG)
               (DETERMINER ((DET1  ONE)))))
```

all the other case strings on this level (major constituents level)* are NIL.

The function GEN_STN_HEAD returns any adverbial indicating time (e.g., the Chinese equivalents of "yesterday", "in 1983", etc.), working on the case string TIME_LOCATION_STR. GEN_SUBJ works on AGENT_STR; GEN_INDOBJ on RECIPIENT_STR; GEN_DOBJ on OBJECT_STR, and so on. Each of these functions check the occurrences of conjunctions, premodifiers and postmodifiers and produce noun-phrases accordingly. The function GEN VERB returns the main verb together with adverbials indicating PLACE_LOCATION, REASON, MANNER, or INSTRUMENT in the order as listed above; all of them precede the verb. This function produces time marker(s) as well; there are five of them in Chinese: LE, ZUO, GUO, JIAN and ZAI. Time marking in Chinese is far less strict than in English (very often additional means are employed to indicate time. A detailed contrastive description of time marking in Chinese and in English is impossible here, though). The function GEN_POST_VERB_MOD takes RESULT_STR or GOAL_STR, and returns adverbials (or adverbial clauses) indicating the result or the goal of the action the verb describes (e.g., "to kill Mary" in "John made a gun to kill Mary").

2.4 Choosing the Words

In most cases, each sense of an English word (as defined in CASSEX's dictionary) has a single Chinese equivalent (a surface Chinese word). Sometimes one English sense has more than one Chinese equivalent, depending on the context. For instance, "wear" in the sense of "to carry or have (a garment, etc.) on one's person as

* In the notation of dependency grammar we adapt, a major constituent of a given sentence is a constituent immediately dominated by the main verb of the sentence.

clothing, ornament, etc." should be translated as "chuan" in "wear clothes, shoes, stockings, etc"; "daih"* in "wear a hat, jewels, glasses, etc."; and "daa" in "wear a tie". I plan to resolve this multi-choice problem by having extra semantic primitives providing finer word-sense discrimination in the dictionary so that, in the semantic representation produced by the analyser, each word-sense will have just one Chinese equivalent. Then, when generating Chinese words, we just extract those equivalents from the bilingual dictionary where each entry is headed by an English word-sense instead of a word.

2.5 Conjunction Reduction

(Ross 67) defines Conjunction Reduction as follows (p.97):

> We propose a rule of Conjunction Reduction which Chomsky-adjoins to the right or the left of the coordinate node a copy of some constituent which occurs in all conjuncts on a right or left branch, respectively, and then deletes the original nodes.

The semantic representations delivered by the CASSEX package are structures with the deleted constituents recovered. For instance, the representation produced for the sentence

(5) The man kicked and threw the ball.

is roughly

```
((CONJUNCTION AND)
 (CLAUSE
   . . . .
     (V  (KICK (....)
               (AGENT ((MAN....)))
               (OBJECT ((BALL1   ....))))))
 (CLAUSE
  ••••
     (V   (THROW1   (....)
               (AGENT  ((MAN....)))
               (OBJECT ((BALL1   ....)))))))
```

In the generation stage, in order to get well-formed Chinese sentences, we must apply the Conjunction Reduction rule. Only forward deletion of the subject in a conjoined clause and of the attribute in a conjoined NP is obligatory in Chinese (i.e., we only Chomsky-adjoin to the left of the coordinate node a copy of the repeated constituent before deleting the original nodes). This is implemented in our generator so that the output for (5) is

```
RENX TIX    LE       QIUX,  REN  LE  QIUX.
man  kick   PARTICLE ball    throw
```

* I use letters to indicate the four tones for Chinese characters: zero - 1st tone; x - 2nd tone; repetition of the first letter of the vowel - 3rd tone; and h - 4th tone. Examples: MA, MAX, MAA, MAH.

3. Conclusion

The CASSEX package and the generator are written in RUTGERS-UCI LISP and implemented on the University of Essex's PDP-10 computer. A couple of dozen of English sentences, all of them containing the conjunction "and" and involving Gapping or Right Node Raising as well as the common cases of coordination, are tested with the program and good quality Chinese sentences are generated (see Appendix). The project is still in the experiment stage, however. More work needs to be done before it becomes a practical English-Chinese MT system.

Acknowledgements

I would like to thank my colleague Dan Fass for his implementation of part of Boguraev's system on the computer at the University of Essex, and for instructive discussions of this paper with him. Any errors are mine.

Appendix: Some of the Sample Translations

1. The man with the telescope and the umbrella kicked the ball.
DAIH WANGHYUUANJINGHHEX SHAAN DE RENXTIXLE QIUX.
withtelescope andumbrella particle mankickparticle ball.

2. The man with the telescope and the umbrella with a handle kicked the ball.
DAIH WANGHYUUANJINGHHEX DAIH BIING DE SHAAN DE RENXTIXLE QIUX.
handle

3. The man with the telescope and the woman kicked the ball.
DAIH WANGHYUUANJINGHHEX DE RENX NUUERENXTIXLE QIUX.
woman

4. The man with the telescope and the woman with the umbrella kicked the ball.
DAIH WANGHYUUANJINGHDE RENX HEX DAIH SHAAN DE NUUERENXTIXLE QIUX.

5. The man with the child and the woman is kicking the ball.
DAIH XI IAOHAIRX HEX NUUERENXDE RENX ZAIH TIXQIUX.
child particle

6. The man with the child and the woman are kicking the ball.
DAIH XIIAOHAIRX DE RENX HEX NUUERENXZAIH TIXQIUX.

7. The man kicked the ball and the child threw the ball.
RENX TIXLE QIUX, XIIAOHARIX REN LE QIUX.

8. The man kicked the ball and the child.
RENX TIXLEX QIUX HEX XIIAOHAIRX.

9. The man kicked the child and the woman the ball.
RENX TIXLEX XIIAOHAIRX, NUUERENX TIX LEX QIUX.

10. The man kicked the ball and the child threw the ball.
RENX TIXLE QIUX, XIIAOHARIX REN LE QIUX.

11. The man kicked and the woman threw the ball.

RENX TIXLEX QIUX, NUUERENX REN LE QIUX.

12. The old man and woman with the child kicked the ball.
GEN XIIAOHAIRX YIHQII DE LAAO RENX HEX NUUERENX TIX LEX QIUX.
with together

13. The man gave the child a ball and the woman an umbrella.
RENX GEE LE XIIAOHAIRX QIUX, GEE LEX NUUERENX SHAAN.

Bibliography

Boguraev, B. K. (1979) *Automatic Resolution* of *Linguistic Ambiguities.* Technical Report No. 11, University of Cambridge Computer Laboratory, Cambridge.

Burton, R.R. (1976) *Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems.* BBN Report No. 3453, Bolt Beranek and Newman, Ind., Cambridge, MA.

Cater, A.W. (1981) *Analysis and Inference for English*, Technical Report No. 19, University of Cambridge Computer Laboratory, Cambridge.

Goldman, N.M. (1975) "Conceptual Generation," in (Schank 75).

Hankamer, J. (1973) "Unacceptable ambiguity," *Linguistic Inquiry*, 4: 17-68.

Huang, X-M. (1983) "Dealing with conjunctions in a machine translation environment," *Proceedings* of *the Association for Computational Linguistics European Chapter Meeting*, Pisa.

Liu, Y-Q, (1981) "The system of intermediate constituents in machine translation from foreign languages into Chinese," paper presented at the Conference on Chinese Language Use, The Australian National University.

Meehan, J. (1976) *The Metanovel: Writing Stories* *by* *Computer,* Research Report No. 74, Department of Computer Science, Yale University.

Ross, J. R. (1967) *Constraints on Variables in Syntax*. Doctoral Dissertation, MIT, Cambridge, Massachusetts. Reproduced by Indiana Univ. Ling. Club, Bloomington, 1968.

Schank, R.C. (ed) (1975) *Conceptual Information Processing*, Amsterdam: North-Holland.

Shapiro, S.C. (1982) "Generalized augmented transition network grammars for generation from semantic networks," *American Journal of Computational Linguistics*, Vol. 8, No.l, 12-25.

Wilks, Y.A. (1975) "Preference Semantics," Keenan (ed), *Formal Semantics of Natural Language*, Cambridge Univ. Press, London.

Winograd, T. (1971) *Understanding Natural Language,* Edinburgh Univ. Press.

Woods, W. A. (1973) "A experimental parsing system for Transition Network Grammar," Rustin, R.(ed), *Natural Language Processing*, Algorithmic Press, N. Y.