[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

Characteristics of the METAL Machine Translation System at Production Stage

John S. White
Siemens Communication Systems

Summary.

The METAL Machine translation system, a joint project of the Linguistic Research Center and Siemens, has been released for use as part of marketed translation systems. The system, which presently translates technical German into English, is an outgrowth of a traditional, generative approach to automatic analysis and synthesis of natural language phenomena carried on at the Linguistics Research Center for many years. In its present manifestation, it is a modular design consisting of purely monolingual lexicons, transfer lexicons, and an augmented phrase structure grammar. The grammar is powerful enough to constrain application, to build new nodes with essential characteristics of their sons and new synthetic information as well, and to perform transformations to re-order, delete, and create constituents. The parser is enhanced to allow application of rules in levels, and eliminating unlikely paths via preferential weightings calculated from lexical and grammatical data. The METAL system, conceived in recent years as destined for implementation, has an orientation to user interface which includes sophisticated text stripping, unfound word handling and reconstitution, and a convenient means of working with the lexicons interactively,

Keywords. Machine translation, augmented phrase structure grammar, natural language processing, parsing.

1. Introduction.

Research in natural language processing strategies in recent years has proven to be of value in the development of such experimental systems as those natural language understanding, database query design, and expert systems. General issues in natural language processing have aided, and have been elucidated by, experimental work in machine translation.

The METAL system, a device for the machine translation of technical texts from one language to another, is one system that has addressed principal issues of modern natural language processing in the context of its implementation into industrial translation environments. Consequently during its development, METAL has had to demonstrate its coverage of a range of linguistic phenomena which a purely experimental model would never encounter.

A version of METAL that translates German into English has now reached market stage. The LITRAS machine translation system, marketed by Computer Gesellschaft Konstanz, West Germany, employs the METAL system in an office workstation package. Other language pairs are under development for METAL, exploiting the inherent modularity of the parser-grammar-lexicon interaction of which METAL is constituted. German-Spanish and English-German directions are presently under development (which will allow English-Spanish as an automatic consequence), and an experimental prototype of German-Chinese has achieved considerable success.

The Linguistics Research Center, at which the METAL system has been developed, began research in machine translation in 1961, funded from various sources such as Air Force Rome Air Development Center. The basic conceptual structure of the METAL grammar was introduced roughly concomitant with the conversion of the program from FORTRAN to LISP in the mid-1970's (Lehmann et al. 1981, Bennett 1983, Slocum 1983). Refinements in underlying linguistic approach have been implemented along with improvements in software, conversion among LISP versions, and improvements in machinery to the present, commercial stage.

This discussion will treat the salient design features of the METAL lexicons, grammar, parser, and the user-oriented enhancements which make the system more amenable to actual needs of the technical translator.

2. Lexical Databases.

The METAL translation run will make use of three lexicons: a source-language lexicon (German, in the present implementation), a target-language lexicon (English), and a transfer lexicon (German-English). Each monolingual lexical entry is a translation-independent set of feature-value pairs relevant to that lexical entry. The set makes available to the grammar such lexical information as inflectional behavior, co-occurrence patterns (e.g., what objects a verb lexical entry might expect), and the actual shape that entry will have in a text to be analyzed or generated.

The transfer lexicon connects one monolingual lexicon with another, relating a lexical entry in one with one or more lexical entries in the other. At the same time, the transfer lexicon can provide information about which lexical translation to choose when there are multiple options, either by constraining or by preferring certain transfers in certain contexts. The constraints are imposed via tests on features which the source or target entry either has natively (from their monolingual entries) or has inherited from the grammar. Additionally, the transfer lexicon allows for calls to powerful functions which can re-organize the arguments identified with a lexical entry (as when, for example, a direct object of a particular verb in the source language must be expressed as the object of a preposition in the target).

3. Grammar Component.

Each of the lexical elements of METAL contains mechanisms of sufficient power to perform operations conditional to co-occurrence of features. Each is theoretically capable of re-ordering, creating new string elements, or deleting. In short, the METAL lexical component is able to handle much more of the translation process than it actually does, virtually to the point of being a purely lexically-driven MT system of the sort described by Garvin (1972).

However, to emulate the linguistic generalizations required to translate large documents would require massive amounts of such process information to reside in the lexicon, with no obvious way to avoid redundancy. Operations could not be readily generalized for the primary word classes which have them in common, let alone for the secondary subclasses resulting from other operations. It is for that reason that the locus of linguistic information in the METAL system, both for analysis and synthesis, is the grammar rule.

The set of grammar rules in METAL forms a phrase structure grammar, which is augmented to constrain application based on constituents and their interaction, pass certain facts up to the superordinate nodes of a phrase structure tree, and re-configure the constituents themselves.

The analytic portion of the grammar rule creates a node when a string of constituents is found which both meets the structural description and passes the tests. In general, the node has the characteristics of the head of the constituent set which it dominates. In this sense there is a common ground with the various X-bar derivative linguistic models (e.g., Gazdar 1982, Karttunen 1984, Kay 1984, Pollard 1984, Kaplan and Bresnan 1983). At appropriate points in the derivation of a constituent, a constituent phrase may be treated as if it were simply the head of that phrase. Typically, for example, the value for CAN (the dictionary-entry form of a word) is copied from the head of a phrase up to the new node. This value may allow the unique transfer of a whole phrase on the basis of that value of CAN. The METAL grammar may differ significantly from such X-bar grammars in that there is no internal prescription toward endowing the created node with the characteristics of its head. METAL has the capability of copying properties of any constituent, or of computed values of constituents, or of externally imposed values. METAL can, and often does, give the node it builds the characteristics of its non-head constituents. In the most general ways, though, the grammar component resembles modern feature-value phrase structure treatments in that it maintains derivational histories of nodes as properties of the nodes.

Figure 3-1 shows a typical METAL grammar rule, one which builds a clause from a prepositional phrase followed by a right-branching clause. This rule is involved in the translation of the examples given below.

Specifically, the grammar rule may have the following characteristics:

A. Tests on constituents. Constituents which meet the description for application are tested for certain properties, in order to constrain application of the rule. In Fig. 3-1, these tests specify, for example, that RCL not be a verb phrase in the imperative mood.

One of the constituent tests is made on the head node itself. This constrains the rule to apply at the "level" specified. The rule will not be tried unless possibly applicable rules of a lower level have tried and failed. The rule in Fig.3-1 is a level three rule, i.e, there must be no successful parse to S along any path using rules of level 1 or 2.

B. Tests among constituents. Properties of constituents are tested against each other to determine whether they can interact in ways predicted by the linguistic information in the lexicon and in the rules themselves. Agreement is tested in this way. Rules will fail to apply if the interaction does not meet the conditions of the test.

In rules such as that in Fig.3-1 involving clauses as constituents, a subroutine is invoked in TEST which determines the subject, objects, and peripheral arguments of a predicate and writes the constituents in a corresponding order. This work is performed in TEST with the call to the FRM subroutine.

A rule may have transformations associated with it. These may be invoked from any portion of the rule. They have the power to rearrange subordinate constituents, delete constituents, add constituents or add features to constituents. The transformation may be written in the rule or called from a set of external subroutines. In the example rule, an external transformation ("CLAUSE2") is called via XFM in TEST. Transformations may be specified internally as well. If the structural description of a transformation is not met, the transformation fails; if a transformation in TEST fails, the rule fails. Consequently, the use of transformations in TEST is a particularly powerful use for them, in that they constrain and manipulate at the same time.

```
CLS         PP                          RCL
0           1                          2
(LVL 3)     (REQ CAN * anstatt ob ohne statt um)  (OPT MD * IMP)
                                       (OR (OPT NOAUX NIL)
                                       (REQ SPX)
                                       (REQ PX NIL))

AUTHOR    "Root on 12/07/84 11:50:49"
TEST
        (OR (LCM $)
            (LCM PNCT)
            (LCM (CONJ:1 NIL (REQ CU COR AJT))))
        (OR (RCM $)
            (RCM PNCT)
            (RCM PAR)
            (RCM (CONJ:1 NIL (REQ CU COR AJT))))
       (XFM CLAUSE2)
       (FRM)
CONSTR
       (AND (RET 2 INT)
       (ADD KI WH)
       (ADD MD Q))
INTEGR
        (RES)
ENGLISH
        (AND (INT 2 DA T)
             (SEV 2 CON T))
        (SEF 1 MD)
        (CLSXFR)
        (ORO)
        (XFR)
---------
((LHS CLS RHS (PP RCL)))
```

Figure 3-1: Typical clause rule in METAL grammar

A recent innovation to the METAL grammar has been the introduction of left-context sensitivity. The rule can examine the strings to the left of the node under consideration, and constrain the application of the rule accordingly. In the example, the rule tests for the presence of sentence markers or certain kinds of punctuation to the left of what will be the CLS if successfully applied.

C. Construction of superordinate node. When a string of constituents passes all tests in an applicable rule, a new entity (node) is built which contains information from the constituents. Some information is copied to the new node, some is left off, and some is created based on the discovered interaction among the constituents. In Fig.3-1, features associated with question forms are synthesized if a feature INT on the RCL is present.

D. Integration of coreferential constituents. Phrase structure trees may be scanned to find the antecedents of pronouns. Such pronouns are then given some of the properties of the antecedent. Anaphora resolution can occur outside as well as inside sentence boundaries. In the translation example below, the antecedent of "seine" and "ihn" is correctly identified by the INTEGR portion of the relevant noun-phrase rule.

E. Transfer into target language. The transfer section of a grammar rule (named after the target language) prepares the constituents of a node for transfer into the target language, and then develop and propagate target-language specific properties up a transferred tree. Information written in this portion of the rule is specific to the target language. Information can be brought down to subordinate constituents from superordinate constituents, or added to subordinate nodes. The call to the function XFR passes control to the transfer section of the rules associated with each of the subordinate nodes, which in turn is passed to the transfer section of their subordinates' rules, and so on until lexical transfer occurs. After this recursive application of XFR, the constituents referred to in the rule have been translated; certain other items of information about the target language can now be sent back up the tree. In Fig.3-1, The ENGLISH portion creates a value for the PP to inherit if it has the feature DA, and causes the RCL to inherit the mood of CLS (note that the reference of numbers to constituents has switched, owing to the fact that their order was switched as part of the grammatical-function framing in CONSTR)

The following example is a translation of a sentence, demonstrating the predicate-argument identification handled by the rule given above, as well as the resolution of the anaphoric pronominal in the subordinate clause.

```
 (translate)
Sentence: (Der Mann befand sich in Muenchen nach dem Krieg, ohne
dass seine Frau ihn gefunden hat)
2 interpretations in 1788 milliseconds:   894 msecs/interp.
169 PHRASES: 119 REJECTED.
Transfer plus generation time: 2917 milliseconds.
(|the   |man| |was| |in| |Munich| |after| |the| |war| |without|
 |his|  |wife| |having| |found| |him| )
```

4. Semantic Handling in METAL.

The overall design of the METAL grammar allows the synthesis and inheritance of features as properties of created nodes, and preserves these properties through the entire translation process. The mechanism already exists, therefore, to enable the inclusion of a semantic interpretation of the rule's right-hand-side as a property. Such an interpretation property of the node could then be involved in the calculation of an interpretation property of nodes which dominate it, resulting ultimately in a sentential semantic interpretation. In this METAL could produce semantic interpretation along the lines of post-Montague mechanisms, i.e., via the association of individual interpretive processes with individual syntactic rules (e.g., Gazdar op.cit, Root 1982, Rosenschein and Shieber 1982). Each interpretation, built compositionally by alogorithms associated with the parsing operation, could be used as strings of an interlingua, thence to be translated directly into the target syntax that represents that interpretation.

This strategy has not been employed for two reasons. First, it is believed that such a strategy is far less efficient computationally, especially in the translation among Indo-European languages, in which syntactic phenomena in the source can be related algorithmically to syntactic phenomena in the target. Such efficiency is, of course, relevant in consideration of industrial implementation. Secondly, METAL already uses devices of a "semo-syntactic" nature to arrive at the "canonical structure" of clauses, transferring these into corresponding canonical structures in the target. These devices identify the grammatical function of predicative constituents, as shown above.

METAL also employs lexical semantics, combining selective information about the reference of nouns and adjectives for use in interpretation, lexical transfer of associated constituents, and de-adjectival derivation. The following example, in which the word "Buch" is substituted for "Krieg" demonstrates how the transfer of the preposition "nach" depends upon the lexical-semantic information inherited from the head of its object noun phrase.

```
Sentence: (Der Mann befand sich in Muenchen nach dem Buch, ohne
dass seine Frau ihn gefunden hat)
1 interpretation in 2040 milliseconds.
150 PHRASES: 107 REJECTED.
Transfer plus generation time: 4021 milliseconds.
(|the| |man|  | was|   |in| |Munich| |according|   |to|   |the|
|book| |without| |his| |wife| |having| |found| |him|)
```

In the previous example, "nach" translated as "after", with "Krieg", since the latter can refer to events in a span of time. In the substitution of "Buch" above, "nach" translated as "according to", corresponding to the tangible characteristics of the possible referents of "Buch".

5. Analysis of compounds.

The ability to break compound German words into smaller, known components has already demonstrated a capability to reduce the amount of lexical coding work required by the human translator. In both the pre-analysis routines and in translation, the compound analyzer can reduce such compounds to known lexical strings. The lexical coder can then determine whether the one-for-one transfer of the compound components is a suitable translation for the term, or whether the term should be entered as monolexemic. Further work will determine whether calculations on semantics of the components can be used to determine modificational scope of the components, and thus their output order in a target language. Of course, this capability is immediately relevant to handling the relations possibly obtaining among English multiple noun strings, for use in determining derivational and ordering requirements for target languages when English is the source. Meanwhile, the enabling of this compound analysis process (it can be turned on or off) can create interesting side effects, as shown when "Darmstadt" is substituted for "Muenchen":

```
(translate)
Sentence: (Der Mann befand sich in Darmstadt nach dem Krieg, ohne
dass seine Frau ihn gefunden hat)
2 interpretations in 1749 milliseconds:   874 msecs/interp.
171 PHRASES: 120 REJECTED.
Transfer plus generation time: 3435 milliseconds.
(|the| |man| |was| |in| |the| |intestine| |city| |after| |the|
|war| |without| |his| |wife| |having| |found| |him|)
```

6. Parsing component.

METAL effects what Slocum (Slocum et al., 1984) has referred to as a "some-paths" parsing strategy through the interaction of a left-corner, bottom-up parsing algorithm and leveling constraints with the grammar rules. The parser applies all the grammar rules up to a predefined level on a text sentence, and stops when at least one S (a successful parse of the entire unit) is achieved. If no S is reached, the level is incremented and the procedure is repeated, including both the old and new rules. In this way an all-paths parsing strategy is optimized to produce the best interpretations along the least number of paths.

The basic parser is optimized to minimize re-computations along each path, by use of a chart parser strategy. It has been augmented further to avoid unnecessary computation by "giving up" after a predetermined number of phrases-per-word has been found. After a certain high number, it is assumed that the cost in efficiency has outweighed the chances that a successful parse might be found in an as yet untried path. In this event, METAL outputs the string of the transfers of the longest successful phrases it found; this "phrasal dump" at least provides the translator with basic constituent structures and accurate terminology for post-editing. Often the phrasal happens to be a useable target sentence.

In conjunction with the strategies for limiting paths, the preference of a phrase, computed from lexical and grammatical preference, is employed to prefer certain paths over others, to prefer one interpretation over others in the event of multiple successful parses of a sentence, and to output the longest and/or best phrases for a phrasal dump.

7. User interface.

The underlying orientation of the recent METAL development effort has been toward a system with sound theoretical underpinnings, yet skewed toward the eventuality that human translators would be using a version of METAL to do translation. Thus at crucial decision points there has been an awareness favoring options that best enabled user applications,

7.1. The Intercoder.

On the LISP machines, a routine exists which allows a user to code lexical entries without directly making database calls. This "intercoder" is an interactive, menu-driven procedure which takes a user's response to questions about aspects of the lexical item being coded, and then creates or updates an entry transparently. Since applications of METAL may differ depending upon the vendor and the need, the Linguistic Research Center model of the is treated as a developmental feature. It is, however, seen as a prototype for the production version of the lexical coding procedure.

7.2. Text handling component.

Other than terminology development, the non-linguistic task of processing on-line text formats is the most time consuming for the user. Purely experimental machine translation systems have not been concerned with such matters as automated handling of text requirements specific to particular word-processors. From the point of view of industrial implementation, however, the benefits of rapid, consistent translation of content can be erased by the task of manually re-formatting the output into a form resembling that of the original source document. The final evaluation of machine translation is, from this perspective, its cost effectiveness. If a system requires the destruction of formats in order to translate, that cost effectiveness is lost and the system as a whole fails, regardless of its linguistic or computation sophistication.

With this awareness of effectiveness as a holistic concept, METAL has developed software for handling text in a way that provides the post-editor with cost-effective options for re-formatting. METAL uses a series of routines to strip sentence units from formatted text, to pre-analyze the text for unknown words and misspellings, and produce list translation output in a format most convenient to the post-editor.

The sentence stripping procedure is a human-aided process in which an automatic program attempts to mark sentence boundaries. The output is corrected by human pre-editing, and further marks are made to delimit, text structures to be left in place but not translated. The result of this process is two files, one containing the original format with addressed references to the sentence unit in each format position, and a file containing the list of sentences to be translated by METAL. These routines can take text from multi-column input, from tables, flow charts, and from text containing word-processor commands.

The pre-analysis routine scans the text and attempts to find the uninflected root of each word, and to look that word up in the lexicon. Failing this, pre-analysis writes this word in an "unknown words" file. Lexicographers may use this file to update the lexicons prior to a translation run.

The post-translation procedure begins with two files, one an interlinear list of the original sentences and their translations, and a file which has written the translated sentences into the original formats of the source text. The human post-editor has the option of correcting either the reconstituted translated text, or correcting the interlinear text and using that as the input to the reconstitution program.

8. Conclusion.

METAL is the first of its generation of linguistic and computational approaches to translation to be brought out into commercial production. It embodies modern natural language tools of both linguistic and computational nature, yet in a way that is intended to be oriented toward naive users. Acceptance of the system in the market will not be solely a result of the sophistication of the translation routines themselves. The usual factors of startup costs, support, etc., will play a role. But the capabilities of the linguistic and computational approaches, along with a usable text support environment, will contribute to the acceptance of the variety of natural language systems employing similar technologies.

9. Acknowledgments.

I would like to thank the staff of the Linguistics Research Center, University of Texas, for their assistance in preparing this paper, and in particular to Rebecca Root for her helpful comments.

References

Bennett, W.S. 1982. The Linguistic Component of METAL. Working Paper LRC-82-2, Linguistics Research Center, University of Texas at Austin.

Garvin, P.L. 1972. On Machine Translation. The Hague: Mouton.

Gazdar, G. 1982. Phrase Structure Grammar. In P.Jacobson and G.Pullum (eds.), The Nature of Syntactic Representation. D.Reidel.

Kaplan, R., and J.Bresnan. 1983. Lexical-functional grammar: a formal system for grammatical representation. in J. Bresnan (ed.) The Mental Representation of Grammatical Relations. MIT Press. Chapter 4.

Karttunen, L. 1984. Features and values. Proceedings of COLING84. pp. 28-33.

Kay, M. 1984. Functional unification grammar: A formalism for machine translation. Proceedings of COLING84. pp 75-79.

Lehmann,W.P., W.S. Bennett, J. Slocum, H. Smith, S.M.V. Pfluger, and S.A.Eveland. 1981. The METAL system. Final technical Report, RADC-TR-80-374, Linguistics Research Center, University of Texas at Austin. NTIS AO-97896.

Pollard, C. 1984. Generalized Phrase Structure Grammars, Head Grammars, and Natural Language. Ph.D. Dissertation, Stanford University.

Rosenschein, S.J., and S.M. Shieber. 1982. Translating English into logical form. Proceedings of the 20th Meeting of the Association for Computational Linguistics. pp 1-8.

Root, R. 1982. Parsing with logical forms. Texas Linguistic Forum 21:179-204.

Slocum, J. 1983. A status report on the LRC machine translation system. ACL Proceedings, Conference on Applied Natural Language Processing, pp.166-173.

Slocum, J., W.S. Bennett, J. Bear, M. Morgan, R. Root. 1984. METAL: The LRC Machine translation system. Working Paper LRC-84-2, Linguistics Research Center, University of Texas at Austin.