A Status Report on the LRC Machine Translation System

Jonathan Slocum

Linguistics Research Center
University of Texas

Working Paper LRC-82-3
December, 1982

This is a very slightly updated version of a paper that was presented at the Conference on Applied Natural Language Processing, which was held on Feb. 1-3, 1983, in Santa Monica, California; it was co-sponsored by the Association for Computational Linguistics and the Naval Research Laboratory.

A STATUS REPORT ON THE LRC MACHINE TRANSLATION SYSTEM

Jonathan Slocum
Linguistics Research Center
University of Texas

ABSTRACT

This paper discusses the linguistic and computational techniques employed in the current version of Machine Translation system being developed at the Linguistics Research Center of the University of Texas, under contract to Siemens AG in Munich, West Germany. We pay particular attention to the reasons for our choice of certain techniques over other candidates, based on both objective and subjective criteria. We then report the system's status vis-a-vis its readiness for application in a production environment, as a means of justifying our claims regarding the practical utility of the methods we espouse.

I INTRODUCTION

The LRC MT system is one of very few large-scale applications of modern computational linguistics techniques [Lehmann, 1981]. Although the LRC MT system is nearing the status of a production system (a version will be delivered to the project sponsor soon after this conference takes place), it is not at all static; rather, it is an evolving collection of techniques which are continually tested through application to moderately large technical manuals ranging from 50 to 200 pages in length. Thus, our "applied" system remains a research vehicle that serves as an excellent testbed for proposed new procedures.

In general, the criteria for our choice of linguistic and computational techniques are three: effectiveness, convenience of use, and efficiency. These criteria are applied in a context where the production of an MT system to be operational in the near-term future is of critical concern. Candidate techniques which do not admit near-term, large-scale application thus suffer an over-whelming disadvantage. The questions confronting us are, then, twofold: (1) which techniques admit such application; and (2) which of these best satisfy our three general criteria? The first question is usually answered through an evaluation of the likely difficulties and requirements for implementation; the second, through empirical results in the course of experiments.

Our evaluation of the LRC MT system's current status will be based on three points: (a) the system's provision of all the tools necessary for users to effect the complete translation process (including text processing, editing, terminology maintenance, dictionary look-up, etc.); (b) quantitative performance (i.e., throughput on a particular target machine; (c) qualitative performance; and (d) what is known about overall performance and cost-effectiveness (i.e., the number of revisors [likely to be] supported by a single "copy" of the system, their [expected] throughput, an accounting of any other personnel necessary for the normal day-to-day operation of the system,

and the [projected] overall costs of translation as it compares to the norm experienced in human translation). The "final numbers" will not be available at the time of the conference, but the results of some preliminary experiments by our sponsor will be in, and thus some reasonable projections can be made.

II LINGUISTIC TECHNIQUES EMPLOYED

Our distinction between "linguistic techniques" and "computational techniques" (discussed in the next major section) is somewhat artificial, but it has some validity in a broad sense, as should become clear from an overview of the points considered. In this section we present the reasons for our use of the following linguistic techniques: (a) a phrase-structure grammar; (b) syntactic features; (c) semantic features; (d) scored interpretations; (e) transformations indexed to specific rules; (f) a transfer component; and (g) attached procedures to effect translation.

A. Phrase-Structure Grammar

In the LRC MT system we employ a phrase-structure grammar, augmented by sufficient lexical controls to make it resemble lexical-functional grammar [Bresnan, 1977]. Of all our linguistic decisions, this is surely the most controversial, and consequently will receive the most attention. Generally speaking, there are two competing claims: first, that syntax rules per se are inadequate and wasteful (e.g., [Cullingford, 1978]); and second, that other forms of grammar (ATNs [Woods, 1970], transformational [Petrick, 1973], procedural [Winograd, 1972], word-experts [Small, 1980], etc.) are superior. We will deal with these in turn.

There are schools of thought that claim that syntax rules per se are inappropriate models of language. Language should, according to this notion, be treated [almost] entirely on the basis of semantics, guided by a strong underlying model of the current situational context, and the expectations that may be derived therefrom. We cannot argue against the claim that semantics is of critical concern in Natural Language Processing. However, as yet no strong case has been advanced for the abandonment of syntax. Moreover, no system has been deleloped by any of the adherents of the "semantics only" school of thought that has more-or-less successfully dealt with ALL of a wide range -- or at least large volume -- of material. A more damaging argument against this school is that every NLP system to date that HAS been applied to large volumes of text (in the attempt to process ALL of it some significant sense) has been based on a strong syntactic model of language (see, e.g., [Boitet et al., 1980b], [Damerau, 1981], [Hendrix et al., 1978], [Lehmann et al., 1981], [Martin et al., 1981], [Robinson, 1982], and [Sager, 1981]).

There are other schools of thought that hold phrase-structure (PS) rules in disrespect, while admitting the utility (necessity) of syntax. It is claimed that the phrase-structure formalism is inadequate, and that other forms of grammar are necessary. (This has been a long-standing position in the linguistic community, being upheld there before most computational linguists jumped on the bandwagon; ironically, this position is now being challenged by some within the linguistic community itself, who are once again supporting PS rules as a model of natural language use [Gazdar, 1981].) The anti-PS positions in the NLP community are all, of necessity, based on

practical considerations, since the models advanced to replace PS rules are formally equivalent in generative power (assuming the PS rules to be augmented, which is always the case in modern NLP systems employing them). But cascaded ATNs [Woods, 1980], for example, are only marginally different from PS rule systems. It is curious to note that only one of the remaining contenders (a transformational grammar [Damerau, 1981]) has been demonstrated in large-scale application -- and even this system employs PS rules in the initial stages of parsing. Other formal systems (e.g., procedural grammars [Winograd, 1972]) have been applied to semantically deep (but linguistically impoverished) domains -- or to excessively limited domains (e.g., Small's [1980] "word expert" parser seems to have encompassed a vocabulary of less than 20 items).

For practical application, it is necessary that a system be able to accumulate grammar rules, and especially lexical items, at a prodigious rate by current NLP standards. The formalisms competing with PS rules and dictionary entries of modest size seem to be universally characterizable as requiring enormous human resources for their implementation in even a moderately large environment. This should not be surprising: it is precisely the claim of these competing methodologies (those that are other than slight variations on PS rules) that language is an exceedingly complex phenomenon, requiring correspondingly complex techniques to model. For "deep understanding" applications, we do not contest this claim. But we do maintain that there are some applications that do not seem to require this level of effort for adequate results in a practical setting. Our particular application -- automated translation of technical texts -- seems to fall in this category.

The LRC MT system is currently equipped with approximately 400 PS rules describing the Source Language (German), and around 10,000 lexical entries in each of two languages (German, and the Target Language -- English). The current state of our coverage of the SL is that the system is able to parse and acceptably translate the majority of sentences in previously-unseen texts, within the subject areas bounded by our dictionary (specific figures will be related below). By the time this conference convenes, we will have begun the process of adding to the system an analysis grammar of the current TL (English), so that the direction of translation may be reversed; we anticipate bringing the English grammar up to the level of the German grammar in about a year's time. Our expectations for eventual coverage are that around 1,000 PS rules will be adequate to account for almost all sentence forms actually encountered in technical texts, whatever the language. We do not feel constrained to account for every possible sentence form in such texts -- nor for sentence forms not found in such texts (as in the case of poetry) -- since the required effort would not be cost-effective whether measured in financial or human terms, even if it were possible using current techniques (which we doubt).

B. Syntactic Features

Our use of syntactic features is relatively noncontroversial, given our choice of the PS rule formalism. We employ syntactic features for two purposes. One is the usual practice of using such features to restrict the application of PS rules (e.g., by enforcing subject-verb number agreement). The other use is perhaps peculiar to our type of application: once an analysis

is achieved, certain syntactic features are employed to control the course (and outcome) of translation -- i.e., generation of the TL sentence. The "augmentations" to our PS rules include procedures written in a formal language (so that our linguists do not have to learn LISP) that manipulate features by restricting their presence, their values if present, etc., and by moving them from node to node in the "parse tree" during the course of the analysis. As is the case with other researchers employing such techniques, we have found this to be an extremely powerful (and of course necessary) means of restricting the activities of the parser.

C. Semantic Features

We employ simple semantic features, as opposed to complex models of the domain. Our reasons are primarily practical. First, they seem sufficient for at least the initial stage of our application. Second, the thought of writing complex models of even one complete technical domain is staggering: the operation and maintenance manuals we are currently working with (describing a digital telephone switching system) are part of a document collection that is expected to comprise some 100,000 pages of text when complete. A research group the size of ours would not even be able to read that volume of material, much less write the "necessary" semantic models subsumed by it, in any reasonable amount of time. (The group would also have to become electronics engineers, in all likelihood.) If such models are indeed required for our application, we will never succeed.

As it turns out, we are doing surprisingly well without such models. In fact, our semantic feature system is not yet being employed to restrict the analysis effort at all; instead, it is used at "transfer time" (described later) to improve the quality of the translations, primarily of prepositions. We look forward to extending the use of semantic features to other parts of speech, and to substantive activity during analysis; but even we were pleased at the results we achieved using only syntactic features.

D. Scored Interpretations

It is a well-known fact that NLP systems tend to produce many readings of their input sentences (unless, of course, constrained to produce the first reading only -- which can result in the "right" interpretation being overlooked). The LRC MT system produces all interpretations of the input "sentence" and assigns each of them a score, or plausibility factor [Robinson, 1982]. This technique can be used, in theory, to select a "best" interpretation from the possible readings of an ambiguous sentence. We base our scores on both lexical and grammatical phenomena -- plus the types of any spelling/typographical errors, which can sometimes be "corrected" in more than one way.

Our experiences relating to the reliability and stability of heuristics based on this technique are decidedly positive: we employ only the (or a) highest-scoring reading for translation (the others being discarded), and our informal experiments indicate that it is rarely true that a better translation results from a lower-scoring analysis. (Surprisingly often, a number of the higher-scoring interpretations will be translated identically. But poorer translations are frequently seen from the lower-scoring interpretations, demonstrating that the technique is indeed effective.)

E. Indexed Transformations

We employ a transformational component, during both the analysis phase and the translation phase. The transformations, however, are indexed to specific syntax rules rather than loosely keyed to syntactic constructs. (Actually, both styles are available, but our linguists have never seen the need or practicality of employing the open-ended variety). It is clearly more efficient to index transformations to specific rules when possible; the import of our findings is that it seems to be unnecessary to have open-ended transformations -- even during analysis, when one might intuitively expect them to be useful.

F. Transfer Component

It is frequently argued that translation should be a process of analyzing the Source Language (SL) into a "deep representation" of some sort, then directly synthesizing the Target Language (TL) (e.g., [Carbonnel, 1978]). We and others [King, 1981] contest this claim -- especially with regard to "similar languages" (e.g., those in the Indo-European family). One objection is based on large-scale, long-term trials of the "deep representation" (in MT, called the "pivot language") technique by the MT group at Grenoble [Boitet, 1980a]. After an enormous investment in time and energy, including experiments with massive amounts of text, it was decided that the development of a suitable pivot language (for use in Russian-French translation) was probably impossible. Another objection is based on practical considerations: since it is not likely that any NLP system will in the foreseeable future become capable of handling unrestricted input -- even in the technical area(s) for which it might be designed -- it is clear that a "fail-soft" technique is necessary. It is not obvious that such is possible in a system based solely on a pivot language; a hybrid system capable of dealing with shallower levels of understanding is necessary in a practical setting. This being the case, it seems better in near-term applications to start off with a system employing a "shallow" but usable level of analysis, and deepen the level of analysis as experience dictates, and theory plus project resources permit.

Our alternative is to have a `transfer´ component which maps "shallow analyses of sentences" in the SL into "shallow analyses of equivalent sentences" in the TL, from which synthesis then takes place. While we and the rest of the NLP community continue to explore the nature of an adequate pivot language (i.e., the nature of deep semantic models and the processing they entail), we can hopefully proceed to construct a usable system capable of progressive enhancement as linguistic theory becomes able to support deeper models.

G. Attached Translation Procedures

Our Transfer procedures (which effect the actual translation of SL into TL) are tightly bound to nodes in the analysis (parse tree) structure [Paxton, 1977]. They are, in effect, suspended procedures -- the same procedures that constructed the corresponding parse tree nodes to begin with. This is to be preferred over a more general, loose association based on syntactic constructs because, aside from its advantage in sheer computational efficiency, it eliminates the possibility that the "wrong" procedure can be applied to a

construct. The only real argument against this technique, as we see it, is based on space considerations: to the extent that different constructs share the same transfer operations, replication of the procedures that implement said operations (and editing effort to modify them) is possible. We have not noticed this to be a problem. For a while, our system load-up procedure searched for duplicates of this nature and eliminated them; however, the gains turned out to be minimal -- different constructs typically do require different operations.

III COMPUTATIONAL TECHNIQUES EMPLOYED

Again, our separation of "linguistic" from "computational" techniques is somewhat artificial, but nevertheless useful. In this section we present the reasons for our use of the following computational techniques: (a) an all-paths, bottom-up parser; (b) associated rule-body procedures; (c) spelling correction; (d) another fail-soft analysis technique; and (e) recursive parsing of parenthetical expressions.

A. All-paths, Bottom-up Parser

Among all our choices of computational techniques, the use of an all-paths, bottom-up parser is probably the most controversial. It also received our greatest experimental scrutiny. We have collected a substantial body of empirical evidence relating to parsing techniques. Since the evidence and conclusions require lengthy discussion, and are presented elsewhere [Slocum, 1981], we will only briefly summarize the results. The evidence indicates that our use of an all-paths bottom-up parser is justified, given the current state of the art in Computational Linguistics. Our reasons are the following: first, the dreaded "exponential explosion" of processing time has not appeared, on the average (and our grammar and test texts are among the largest in the world), but instead processing time appears to be linear with sentence length -- even though our system produces all possible readings; second, top-down parsing methods suffer inherent disadvantages in efficiency, and bottom-up parsers can be and have been augmented with "top-down filtering" to restrict the syntax rules applied to those that an all-paths top-down parser would apply; third, it is difficult to persuade a top-down parser to continue the analysis effort to the end of the sentence, when it blocks somewhere in the middle -- which makes the implementation of "fail-soft" techniques having production utility that much more difficult; and lastly, the lack of any strong notion of how to construct a "best-path" parser, coupled with the raw speed of well-implemented parsers, implies that an all-paths parser which scores interpretations and can continue the analysis to the end of the sentence may be best in a contemporary application such as ours.

B. Associated Rule-body Procedures

We associate a procedure directly with each individual syntax rule, and evaluate it as soon as the parser determines the rule to be (seemingly) applicable [Pratt, 1973; Hendrix, 1978] -- hence the term "rule-body procedure". This practice is equivalent to what is done in ATN systems. From the linguist's point of view, the contents of our rule-body procedures appear to constitute a formal language dealing with syntactic and semantic

features/values of nodes in the tree -- i.e., no knowledge of LISP is necessary to code effective procedures. Since these procedures are compiled into LISP, all the power of LISP is available as necessary. The chief linguist on our project, who has a vague knowledge of LISP, has employed OR and AND operators to a significant extent (we didn't bother to include them in the specifications of the formal language, though we obviously could have), and on rare occasions has resorted to using COND. No other calls to true LISP functions (as opposed to our formal operators, which are few and typically quite primitive) have seemed necessary, nor has this capability been requested, to date. The power of our rule-body procedures seems to lie in the choice of features/values that decorate the nodes, rather than the processing capabilities of the procedures themselves.

C. Spelling Correction

There are limitations and dangers to spelling correction in general, but we have found it to be an indispensable component of an applied system. People do make spelling and typographical errors, as is well known; even in "polished" documents they appear with surprising frequency (about every page or two, in our experience). Arguments by LISP programmers [re: INTERLISP's DWIM] aside, users of applied NLP systems distinctly dislike being confronted with requests for clarification -- or, worse, unnecessary failure -- in lieu of automated spelling correction. Spelling correction, therefore, is necessary.

Luckily, almost all such errors are treatable with simple techniques: single-letter additions, omissions, and mistakes, plus two- or three-letter transpositions account for almost all mistakes. Unfortunately, it is not infrequently the case that there is more than one way to "correct" a mistake (i.e., resulting in different corrected versions). Even a human cannot always determine the correct form in isolation, and for NLP systems it is even more difficult. There is yet another problem with automatic spelling correction: how much to correct. Given unlimited rein, any word can be "corrected" to any other. Clearly there must be limits, but what are they?

Our informal findings concerning how much one may safely "correct" in an application such as ours are these: the few errors that simple techniques have not handled are almost always bizarre (e.g., repeated syllables or larger portions of words) or highly unusual (e.g., blanks inserted within words); correction of more than a single error in a word is dangerous (it is better to treat the word as unknown, hence a noun); and "correction" of errors which have converted one word into another (valid in isolation) should not be tried.

D. Fail-soft Grammatical Analysis

In the event of failure to achieve a comprehensive analysis of the sentence, a system such as ours -- which is to be applied to hundreds of thousands of pages of text -- cannot indulge in the luxury of simply replying with an error message stating that the sentence cannot be interpreted. Such behavior is a significant problem, one which the NLP community has failed to come to grips with in any coherent fashion. There have, at least, been some forays. Weishedel and Black [1980] discuss techniques for interacting with the linguist/developer to identify insufficiencies in the grammar. This is

fine for development purposes. But, of course, in an applied system the user will be neither the developer nor a linguist, so this approach has no value in the field. Hayes and Mouradian [1981] discuss ways of allowing the parser to cope with ungrammatical utterances; such work is in its infancy, but it is stimulating nonetheless. We look forward to experimenting with similar techniques in our system.

What we require now, however, is a means of dealing with "ungrammatical" input (whether through the human's error or the shortcomings of our own rules) that is highly efficient, sufficiently general to account for a large, unknown range of such errors on its first and subsequent outings, and which can be implemented in a short period of time. We found just such a technique three years ago: a special procedure (invoked when the analysis effort has been carried through to the end of the sentence) searches through the parser's chart to find the shortest path from one end to the other; this path represents the fewest, longest-spanning phrases which were constructed during the analysis. Ties are broken by use of the standard scoring mechanism that provides each phrase in the analysis with a score, or plausibility measure (discussed earlier). We call this procedure `phrasal analysis'.

Our phrasal analysis technique has proven to be useful for both the developers and the end-user, in our application: the system translates each phrase individually, when a comprehensive sentence analysis is not available. The linguists use the results to pin-point missing (or faulty) rules. The users (who are professional translators, editing the MT system's output) have available the best translation possible under the circumstances, rather than no usable output of any kind. To our knowledge, no other NLP system relies on a such a general technique for searching the parser's chart when an analysis effort has failed. We think that phrasal analysis -- which is simple and independent of both language and grammar -- could be useful in other applications of NLP technology, such as natural language interfaces to databases.

E. Recursive Parsing of Parenthetical Expressions

Few NLP systems have ever dealt with parenthetical expressions; but MT researchers know well that these constructs appear in abundance in technical texts. We deal with this phenomenon in the following way: rather than treating parentheses as lexical items, we make use of LISP's natural treatment of them as list delimiters, and treat the resulting sublists as individual "words" in the sentence; these "words" are "lexically analyzed" via recursive calls to the parser. Aside from the elegance of the treatment, this has the advantage that "ungrammatical" parenthetical expressions may undergo phrasal analysis and thus become single-phrase entities as far as the analysis of the encompassing sentence is concerned; thus, ungrammatical parenthetical expressions need not result in ungrammatical (hence poorly handled) sentences.

IV CURRENT STATUS

A. Adequate Support Tools

No NLP system is likely to to be successful in isolation: an environment of support tools is necessary for ultimate acceptance on the part of

prospective users. The following support tools, we think, constitute a minimum workable environment for both development and use of NLP systems: a DBMS for handling lexical entries; validation programs that verify the admissability of all linguistic rules (grammar, lexicons, transformations, etc.) using a set of formal specifications; dictionary programs that search through large numbers of proposed new lexical entries (words, in all relevant languages) to determine which entries are actually new, and which appear to replicate existing entries; defaulting programs that "code" new lexical entries in the NLP system's chosen formalism automatically, given only the root forms of the words and their categories, using empirically determined best guesses based on the available dictionary database entries plus whatever orthographic information is available in the root forms; and benchmark programs to test the integrity of the NLP system after modifications [Slocum, 1982]. A DBMS for handling grammar rules is also a good idea.

For Machine Translation applications, one must add: a collection of text-processing programs that [semi-]automatically mark and extract translatable segments of text from large documents, and which automatically insert translations produced by the MT system back into the original document, preserving all formatting conventions such as tables of contents, section headings, paragraphs, multi-column tables, flowcharts, figure labels, and the like; a powerful on-line editing program with special capabilities (such as single-keystroke commands to look up words in on-line dictionaries) in addition to the normal editing commands (almost all of which should be invokable with a single keystroke); and also, perhaps, (access to) a "term databank," i.e., an on-line database of technical terms used in the subject area(s) to be covered by the MT system.

The LRC MT system already provides all of the tools mentioned above, with the exception of the text editor and terminology database (both of which our sponsor will provide). All of this comes in a single integrated working environment, so that our linguists and lexicographers can implement changes and test them immediately for their effects on translation quality, and modify or delete their additions with ease, if desired.

B. Quantitative Performance

The average performance of the LRC MT system when translating technical manuals from German into English, running in compiled INTERLISP on a DEC 2060 having over a million words of physical memory, has been measured at slightly under 2 seconds of CPU time per input word; this includes storage management (the garbage collector alone consumes 45% of all CPU time on this limited-address-space machine), paging, swapping, and I/O -- that is, all forms of overhead. Our experience on the 2060 involved the translation of some 330 pages of text, in three segments, over a two year period.

On our Symbolics LM-2 Lisp Machine, with 256K words of physical memory, preliminary measurements indicate an average performance of 10-12 seconds (real time) per input word, likewise including all forms of overhead. Our LM-2 experience to date has involved the translation of about 310 pages of text. The paging rate indicates that, with added memory (512K words is "standard" on these machines), we could expect a significant reduction in this performance figure. With a faster, second-generation Lisp Machine, we would expect a more substantial reduction of real-time processing requirements. We

hope to have had the opportunity to conduct an experiment on at least one such machine, by the time this conference convenes.

C. Qualitative Performance

Measuring MT system throughput is one thing. Measuring "machine translation quality" is quite another, since the standards for measurement (and for interpreting the measurements) are little understood, and vary widely. Thus, "quality" measurements are of little validity. However, because there is usually a considerable amount of lay interest in such numbers, we shall endeavor to indicate why they are basically meaningless, and then report our findings for the benefit of those who feel a need to know.

Certainly it is the case that "correctness" numbers can theoretically give some indication of the quality of translation. If an MT system were said to translate, say, 10% of its input correctly, no one would be likely to consider it usable. The trouble is, quoted figures almost universally hover at the opposite extreme of the spectrum -- around 90% -- for MT systems that vary remarkably w.r.t. the subjective quality of their output. (Since, to the lay person, "90% correct" seems to constitute minimal acceptable quality, the consistent use of the 90% figure should not be surprising.)

The trouble arises from at least the following human variables: who performs the measurement? what, exactly, is measured? and by what standards? Since almost all measurements are performed by the vendor of the system in question, there is obvious room for bias. Second, if one measures "words translated correctly," whatever that means, that is a very different thing from measuring, e.g., "sentences translated correctly," whatever that means. Finally, there is the matter of defining the operative word, `correct´. Since no two translators are likely to agree on what constitutes a "correct" translation -- to say nothing of establishing a rigorous, objective standard -- the notion of `correctness´ will naturally vary depending on who determines it. It will also vary depending on the amount of time available to perform the measurement: it is widely recognized that an editor will change more in a given translation, the more time he has to work on it. Finally, `correctness´ will vary depending on the use to which the translation is intended to be put, the classical first division being information acquisition vs. dissemination.

There are a few subsidiary qualifications that must be applied to statements of measured quality: what kinds of text were involved? who chose them? did the vendor have access to them before the test? if so, in what form? and for how long? These are critically important questions relating to the interpretation of the results. It stands to reason that, to get the most trustworthy figures: the system should be applied to such varieties of text as it is intended to handle (in the near term, at least); the texts should be chosen by the user, and not divulged to the vendor beforehand except perhaps in the form of a list of words or technical terms (in root form) which appear therein -- and that, for not too long a period of time before the test.

With the reader bearing all of the above in mind, we report the following quality measurements: during the last two years, LRC personnel have measured the quality of translations produced by the LRC MT system in terms of the percentage of sentences (actually, `translation units´, since isolated words and phrases appear frequently) which were translated from German into

acceptable English; if any change to the translated unit was necessary, however slight, the translation was considered incorrect; the test runs were made once or twice for each text -- once, before the text was ever seen by the LRC staff (a `blind' run), and once more, after a few months of system enhancement based in part on the previous results (a `follow-up' run); the project sponsor always provided the LRC with a list of the words and technical terms said to be employed in the text (the list was sometimes incomplete, as one would expect of human compilations of the vocabulary in a large document). The first run, on a 50-page text, was performed only after the text had been studied for some time; the second and third runs, on an 80-page text, were performed both ways (`blind' and `follow-up'); the fourth test was a blind run on a 200-page text. The figures so measured varied from 55% to 85% depending on the text, and on whether the test was a blind or follow-up run. A fifth test -- a follow-up run on the text used in the fourth test -- has already been performed, but the qualitative results are not available at this writing. The results of this run and two more blind runs on two very different texts totalling 160 pages should be available when the conference convenes; these qualitative results are all to be measured by professional technical translators employed by the project sponsor.

D. Interpretation of the Results

Any positive conclusions we might draw based on such data will be subject to certain objections. It has been argued that, unless an MT system constitutes an almost perfect translator, it will be useless in any practical setting [Kay, 1980]. As we interpret it, the argument proceeds something like this:

(1) there are classical problems in Computational Linguistics that remain unsolved to this day (e.g., anaphora, quantifiers, conjunctions);

(2) these problems will, in any practical setting, compound on one another so as to result in a very low probability that any given sentence will be correctly translated;

(3) it is not in principle possible for a system suffering from malady (1) above to reliably identify and mark its probable errors;

(4) if the human post-editor has to check every sentence to determine if it has been correctly translated, then the translation is useless.

We accept claims (1) and (3) without question. We consider claim (2) to be a matter for empirical validation -- surely not a very controversial contention. As it happens, the substantial body of empirical evidence gathered by the LRC to date refutes this claim. By the time the conference convenes, we will have more definitive data to present, derived by the project sponsor.

Regarding (4), we embrace the assumption that a human post-editor will have to check the entire translation, sentence-by-sentence; but we argue that Kay's conclusion ("then the translation is useless") is again properly a matter for empirical validation. Meanwhile, we are operating under the assumption that this conclusion is patently false -- after all, where translation is taken seriously, human translations are routinely edited via exhaustive review, but no one claims that they are therefore useless!

E. Overall Performance

In this section we advance a meaningful, more-or-less objective metric by which any MT system can and should be judged: overall (man/machine) translation performance. The idea is simple. The MT system must achieve two simultaneous goals: first, the system's output must be acceptable to the translator/editor for the purpose of revision; second, the cost of the total effort (including amortization and maintenance of the hardware and software) must be less than the current alternative for like material -- human translation followed by post-editing.

There may be a significant problem with the reliability of human revisors' judgements (which are nevertheless the best available): the writer has been told by professional technical editors/translators (potential users of the LRC MT system) that they look forward to editing our machine translations "because the machine doesn't care" [private communication]. That is, they would change more in a machine translation than in a supposedly equivalent human translation because they would not have to worry about insulting the original translator with what s/he might consider "petty" changes. Thus, the "correctness" standards to be applied to MT may very possibly differ from those applied to human translation, simply due to the translation source. Since the errors committed by an MT system seldom resemble errors made by human translators, the possibility of a "Turing test" for an MT system does not exist at the current time.

We now have some preliminary data bearing on the issue of overall performance using the LRC MT system. A 70-page text translated by METAL in 30 hours (in a `blind' run) was sent to the project sponsor for revision by a professional staff translator; the revision rate was reported to be 12.85 pages/day. These figures compare favorably to the human translation rate of 4-6 pages/day, and revision rate of 8-10 pages/day, for like text. It can be argued, therefore, that our system may be ready for use in a production translation environment.

V DISCUSSION

We have commented on the relative merits in large-scale application of several linguistic techniques: (a) a phrase-structure grammar; (b) syntactic features; (c) semantic features; (d) scored interpretations; (e) transformations indexed to specific rules; (f) a transfer component; and (g) attached procedures to effect translation. We also have presented our findings concerning the practical merits of several computational techniques: (a) a bottom-up, all-paths parser; (b) associated rule-body procedures; (c) spelling correction; (d) chart searching in case of analysis failures; and (e) recursive parsing of parenthetical expressions. We believe these findings constitute useful information about the state of the art in Computational Linguistics.

We will not have any firm empirical evidence concerning overall performance until later in 1983, when the LRC MT system will have been used in-house by our sponsor, for very-large-scale translation experiments. However, we have presented some preliminary data from our sponsor that can be adduced as a basis for extrapolation. These findings lend credence to our claims regarding the practical utility of the methods we employ.

VI REFERENCES

Boitet, Ch., P. Chatelin, and P. Daun Fraga. "Present and Future Paradigms in the Automatized Translation of Languages," Proc. COLING 80, Tokyo, 1980[a].

Boitet, Ch., and N. Nedobejkine, "Russian-French at GETA: Outline of the Method and Detailed Example," Proceedings of the Eighth International Conference on Computational Linguistics, Tokyo, Sept. 30 - Oct. 4, 1980[b].

Bresnan, J. W., "A Realistic Transformational Grammar," in Halle, Bresnan, and Miller (eds.), Linguistic Theory and Psychological Reality. MIT Press, 1977.

Carbonnel, J., R. E. Cullingford, and A. V. Gershman, "Knowledge-Based Machine Translation," Research Report #146, CS Dept., Yale University, Dec. 1978.

Cullingford, R. E., "Script Application: Computer Understanding of Newspaper Stories," Research Report #116, CS Dept., Yale University, 1978.

Damerau, F. J., "Operating Statistics for the Transformational Question Answering System," AJCL 7 (1), January-March 1981, pp. 30-42.

Gazdar, G., "Unbounded Dependencies and Coordinate Structure," in Linguistic Inquiry, 12 (2), Spring 1981, pp. 155-184.

Hayes, P. J., and G. V. Mouradian, "Flexible Parsing," AJCL, 7 (4), October-December 1981, pp. 232-242.

Hendrix, G. G., et al., "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems 3 (2), June 1978, pp. 105-147.

Kay, M., "The Proper Place of Men and Machines in Language and Translation," Technical Report, Xerox PARC, Palo Alto, California, 1980.

King, M., "Design Characteristics of a Machine Translation System," Proc. 7th IJCAI, Vancouver, B.C., Canada, Aug. 1981, v. 1, pp. 43-46.

Lehmann, W. P., W. S. Bennett, J. Slocum, et al., "The METAL System," Final Technical Report RADC-TR-80-374, Rome Air Development Center, January 1981.

Martin, W. A., K. W. Church, and R. S. Patil, "Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and experimental results," paper presented at the University of Texas Symposium on Modelling Human Parsing Strategies, 24-26 March 1981.

Paxton, W. H., "A Framework for Speech Understanding," Tech. Note 142, AI Center, SRI International, Menlo Park, California, June 1977.

Petrick, S. R., "Transformational Analysis," in R. Rustin (ed.), Natural Language Processing. Algorithmics Press, New York, 1973, pp. 27-41.

Pratt, V. R., "A Linguistics Oriented Programming Language," Proc. 3rd IJCAI, Stanford University, California, August 1973, pp. 372-381.

Robinson, J. J., "DIAGRAM: A Grammar for Dialogues," CACM 25 (1), Jan. 1982.

Sager, N. Natural Language Information Processing. Addison-Wesley, Reading, Massachusetts, 1981.

Slocum, J., "A Practical Comparison of Parsing Strategies for Machine Translation and Other Natural Language Processing Purposes," Tech. Report NL-41, Department of Computer Sciences, University of Texas, August 1981.

Slocum, J., "The LRC Machine Translation System: An Application of State-of-the-Art Text and Natural Language Processing Techniques," COLING 82, Prague, Czechoslovakia, July 5-10, 1982.

Small, S., "Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding," Tech. Rep. 954, CS Dept., Univ. of Maryland, 1980.

Weischedel, R. M., and J. E. Black, "If the Parser Fails," Proceedings of the 18th Annual Meeting of the ACL, Univ. of Pennsylvania, June 19-22, 1980.

Winograd, T. Understanding Natural Language. Academic Press, New York, 1972.

Woods, W. A., "Transition Network Grammars for Natural Language Analysis," CACM, 13 (10), October 1970, pp. 591-606.

Woods, W. A., "Syntax, Semantics, and Speech," BBN Report 3067, Bolt, Beranek, and Newman, Inc., Cambridge, Massachusetts, April 1975.

Woods, W. A., "Cascaded ATN Grammars," AJCL, 6 (1), January-March 1980, pp. 1-12.