The Logos System

Bernard E. SCOTT
Logos Corporation
111 Howard Blvd.
Mt. Arlington, NJ 07856 USA

General Remarks

Logos Corporation has been engaged in the development of machine translation technology for 20 years. From the efforts of these years has emerged the current Logos System, a mature, general-purpose multi-lingual production system installed in over 40 sites in North America and Europe. Source languages in production today are German, with English, French and Italian as targets, and English with German, French and Spanish as targets. Development of an English-Italian system is currently underway in Italy. The system is operational in IBM VM, MVS, in Wang VS and under UNIX on UNYSIS platforms.

Overview of Development History and Motivations

Our company was formed in 1969 in the shadow of the ALPAC report, at a time when the conventional wisdom was that MT could not be done. What motivated our venture was an "idea" of a another way to do MT. Our young company got its first opportunity to try its idea when the US DOD, despite ALPAC, looked to us for help in translating thousands of military manuals into Vietnamese. The English-Vietnamese system we built for the DOD in 1971 was primitive by today's standards but it was successful as far as its intended purpose went. In its brief year of existence as a production system, it translated several million words of technical English into Vietnamese. The DOD's subsequent statement that this E-V system had "established the feasibility of large-scale MT" provided encouragement to our enterprise, but unfortunately did little to promote further government interest in MT. This early system was built in under two years, which shows that it is relatively easy to build a marginal MT system that under the right circumstances can be quite useful. The subsequent long history of our Company, however, also illustrates the corollary, namely, how difficult it is indeed to progress beyond that initial stage.

1973-1981 Development of the Basic Technology

The effort to push back the limits of MT comprises the next period of our company's history. We looked at countless diagnostics of mis-analyzed sentences and tried to discover what the machine needed to "know" in order to get the analysis right This work went on for several years in a process that must be described as entirely inductive. We looked to natural language itself rather than to any linguistic theory or model for our answers. If we had a model at all, it was the psychological model that provided our main point of reference.

It was commonplace in those days to speak of a "semantic wall" that existing machine translation systems were up against. Of course the wall was ambiguity; but disambiguation required semantics, and that was what was missing. No one quite knew how to go about the processing of meaning, at least not in a way that yielded practical results. However, to our way of thinking at Logos, the ambiguity of natural language was not the ultimate show-stopper. Assuming you have found an effective way to provide the necessary semantics, another property of natural language must still be dealt with, one that seemed to pose an even more formidable barrier. I refer to complexity, a condition some describe as situated somewhere midway between order and chaos. To our way of thinking, complexity poses the greater problem for a machine environment. I do not refer here to computational complexity--the machine-time factor required to solve certain classes of problems, nor to the program-space factor, I refer rather to that special kind of complexity proper to natural language processing, what might be termed "cognitive complexity", where the issue is the complexity of the logic and the difficulty the human mind has in maintaining this logic, in simply keeping track of it. It is a problem that only appears late in the development cycle, when things begin to stall. The phenomenon of declining improvement-to-degradation ratios is the all too familiar symptom of cognitive complexity. But whether complexity or ambiguity is seen as the greater problem is incidental--the fact is the two problems together, in their interaction in a machine environment, collectively pose a seemingly insurmountable barrier to success. For example, as you gain on the problem of ambiguity by adding more semantic information, you inevitably loose ground in the struggle against complexity. A case in point is what happens to an ATN when semantics is added. If on the other hand you try to relieve the problem of

complexity by restricting information or the size of the rule base, you weaken the disambiguating power of the system. A case in point here is the paradox of the universal bias toward small rule bases. As the principal means of reducing computational complexity, this bias is easy to understand. But, one asks, how shall a small rule base account for even a modest fraction of the roughly 100,000 context-dependent transformations called for in the average bi-lingual dictionary. We have here the horns of a true dilemma. I might note that the solution of so-called "radical lexicalism" does not seem to solve the dilemma, only to shift it. For one thing, the complexity is shifted to the lexicographer, placing that function well beyond the skills of the average user. For another, generic transformations could not be driven by a dictionary. Finally, there are many dependencies that, because of their subtlety or complexity, appear to be inappropriate for treatment at the dictionary level.

It was the problems associated with this interaction of ambiguity and complexity that we specifically sought to solve as we developed the Logos Model and the semantico-syntactic representation language known as SAL upon which this Logos Model is based. During this period, the Logos Model and SAL were repeatedly refined over a series of prototype systems entailing English, French, Russian, Farsi and German.

1982-1987 Development of a Product

This period witnessed a shift from basic technology to a concern with building a product. Basic linguistic issues like coordination and scoping, etc. continued to receive attention, but a whole new class of problems came to the foreground. The major concerns during this period were:

1. Format command language translation. This Babel of a translation problem within a translation problem was addressed principally via the DCA standard, enabling the system to translate the more common command languages into the target text

2. User-friendly interfaces. A dictionary interface called ALEX was developed to make dictionary work easy for the user. ALEX is a kind of expert system. For example, it automatically generates stems and morphological classifications. ALEX also automates the encoding of entries. For example, ALEX knows all the semantic nuances of all head nouns of a given source language. When the user enters a noun phrase, ALEX displays the set of nuances associated with its head and asks for the user's selection. If the nuance list does not satisfy the user, he or she is invited to suggest a synonym and the nuance list for that synonym is presented. Underlying this strategy is the simplifying principle of encoding by analogy, a practice designed to protect the user from the internals of the system. ALEX also frees the user from having to think about homographs, words that might overlap with an entry even within the same word class. For example, a user entering the word "die Kohle" (coal) will not need to be concerned with the fact that this word overlaps with the plural form of "der Kohl" (cabbage). ALEX does that for the user, behind the scenes. Users can work with ALEX after one or two days training and have no difficulty doing 300 entries in an average work day.

Another user-interface called SEMANTHA was developed to enable the user to add rules to the semantic rule base (known as the semantic table (SEMTAB)). Such rules typically provide for the nuancing of verbal elements in specified contexts but may also address nuancing of other forms. (The semantic table also serves the parse by recognizing dependencies.) This is a particularly interesting interface in that it allows the user "conceptual" access to the deepest levels of analysis. Rules entered through SEMANTHA are in essence "conceptual rules" with no necessary reference to part-of-speech. Thus the same conceptual or deep-structural rule can deal with a variety of surface structures. For example, the default transfer for the verb "aufheben" is "pick up". A user of German-English wishing to transfer "aufheben" as "lift" in the context of words like "Ausnahmezustand" (state of emergency) would enter the following rule in SEMANTHA:

User's Rule: den Ausnahmezustand/etc. aufheben = lift + Object

This single, user-derived "conceptual" rule was responsible for the following translations:

Die Regierung hob den Ausnahmezustand auf.
The government lifted the state of emergency.
Der vorher von der Regierung aufgehobene Ausnahmezustand....
The state of emergency previously lifted, by the government....
Mil Aufhebung des Ausnahmezustands hat die Regierung....
In lifting the state of emergency, the Government....

3. Translators' Workstation. A translators' workstation known as FILIUS (Fully Integrated Language-Independent User System) was developed to provide a window-based interface with all Logos System functions. It also integrates the output of the Logos System within the Nota Bene word processing system for post-editing purposes. A popular feature is multi-window coordinated scrolling between source and

target files. A revision processor is also planned for the FILIUS environment in the near future.

4. Additional Environments. The system product was ported to MVS, VM, WANG VS and to UNIX;

5. Multi-linguality. The Logos Model was extended to a near approximation of a multi-lingual system. By this we mean that: (i) The software of the system is entirely (99.99%) language-independent and is physically shared by all sources and targets. Exceptions relate to such things as compound-noun decomposition (German source) and ellision (French target); (ii) All of the source-language productions and lexicons relating to the analysis of the source sentence are shared by all targets. Indeed, there is only one source analysis path for a given source language sentence, regardless of the target; (iii) Most (but not yet all) of the key components of any given target language's generative rule base are shared by all sources. There is also of course a transfer component in both the rule base and in the lexicon that is language-pair specific.

6. Extensions to SAL-P. This period also saw the strengthening of the procedural language within SAL (SAL-P), which permitted the introduction of a powerful new deterministic English parsing strategy called the Clausal State Parse (CSP). This parse is accomplished entirely by SAL production rules within the RES program shells (Fig. 1) during English source analysis. The Clausal State Parse achieves a 98% success rate in homograph resolution in English source analysis on randomly selected text.

1988-1989 The System Comes of Age

The Logos System in the past few years has emerged from a beta-site posture to full production status, and is presently installed in over 40 systems in Europe and North America. To indicate how the system is being used, some of the recent developments are described. I cite first the selection of the Logos System this year by the Canadian Government after a two year period of evaluation. The Logos System will serve as the general-purpose machine translation component in the Government's long-range plans for a network-based government-wide automation of the translation process. At present there are four installation sites, with additional sites to follow.

Also in Canada there was formed last year an interesting new translation company that has predicated its entire translation effort on machine translation. This company employs 30 full-time translators who work mostly on Canadian Navy shipbuilding projects. The translation setup entails a local area network connecting several VAX servers, a Kurzweil OCR, a graphics scanner, 30 Microvax/Interleaf workstations and an IBM 9370 on which is housed the Logos System. In the past six months, this company has entered 160,000 nautical engineering terms into the Logos System dictionary.

Another interesting development this year was the selection of the Logos System by the Deutsche Bank for use in its interoffice communications network. The Logos System will be used primarily as an information utility on the network where most of the translations will be delivered unedited. Tests by the Deutsche Bank reveal that unedited raw translations of interoffice memos and letters achieve an understandability and reliability index of 96%.

In another new development, Logos has been invited to form a strategic business relationship with a large multi-national computer vendor who will be licensed to market the Logos System worldwide under its own Logo as an offering in its office automation product line. The Logos System was selected after two years of product evaluation in both Europe and North America where it was determined that "substantial reductions in translation costs can be effected with the Logos System".

These cases are cited as evidence that general-purpose machine translation of unconstrained text has unarguably come of age.

The Logos Model

Natural language translation is often thought of as a mapping problem, mapping a sentence in one natural language into something equivalent in another natural language. Now it's perfectly obvious to those of us who work in this field that a machine cannot do this on natural language's own terms; a machine cannot manipulate the 100,000-500,000 symbols which constitute a given language set, not in any meaningful way. What a machine must first do is reduce natural language to an artificial or formal language. Then it has a chance of succeeding. While this may seem all fairly obvious, theoreticians appear to have left it to linguistic

engineers to figure out how this is to be done. How this is to be accomplished constitutes the heart of our approach. Our strategy is straightforward: take away from natural language that which differentiates natural language from formal or artificial language, namely, ambiguity and complexity. Formal or artificial language may be thought of as natural language bereft of these two properties. It follows from this that to the extent one can disambiguatc and decomplexify, one can solve the problem of machine translation. We believe the obverse is also true.

The Logos Model shown in Fig. 1 is an elaborate multi-phased mechanical disambiguator and de-complexifier, a pipeline in the course of which disambiguation and decomplexification take place incrementally, in stages. The terminus is a formal, abstract, semantico-syntactic representation of the natural language string that can then be mapped into a target language with relative ease. The Model relies on the power of SAL, the semantico-syntactic abstraction language into which the natural language stream is re-expressed and on the all-important fact that the rule base is also written in SAL.

Semantico-Syntactic Abstraction Language (SAL)

I stated earlier that the technological underpinnings of the Logos Model were developed inductively. I also said that we tried to model the system on the psychological model as best we understood it. Finally, I suggested that we had an "idea" of how to approach machine translation in an entirely new way. Let me now try to explain how all this came together in SAL and in the Logos Model's employment of SAL.

Working inductively, we arrived at the conclusion that structure, at least deep structure, is a function of meaning. This, it seems to me, is the underlying principle behind case and valency grammars. We at Logos arrived at something akin to a valency grammar by simple induction, by looking at what the machine needed to know in order to parse correctly and discovering the clue to be in the semantics of the verbal element. We also saw that verbal elements or ingredients are found in all parts of speech. For example, the noun "way" has a verbal coloration or bias which, when encoded as such, enables a machine to differentiate between "ways of revolving credit" and "types of revolving credit". A similar statement can be made about the adjectives "easy" and "eager". These kinds of properties of course are generally learned by anyone who works in this field. What we did was to use them to create subclassifications within each part of speech, and then to elaborate all this into a full-blown, hierarchical, semantico-syntactic abstraction language called SAL. It is an abstraction language because it deals with second-order concepts. It is semantico-syntactic because semantics and syntax are integrated, are seen in fact as the two extremes of a continuum, and because the particular semantic property we focused on was that conceptual property that had syntactic effect elsewhere, that in effect created dependencies. Finally, SAL is a language and not a set of properties, in that natural language can be mapped into SAL and then thereafter be handled in terms of its SAL representation. In so doing, there is a reduction in semantic complexity of two orders of magnitude, from a 100,000 element set to a 1000 element set, the approximate size of the SAL vocabulary set. In this you will recognize the attempt to supply semantics so as to deal with disambiguation without at the same time exacerbating the problem of complexity.

I said earlier that we attempted to imitate the psychological model as best we understood it. It is widely recognized, for example, that the mind abhors complexity and that its principal means for avoiding it is abstraction, by assimilating differences to something common and therefore more abstract In much the same way, SAL achieves semantic simplification through abstraction, through the reduction of intensional values to second-order concepts. The analysis performed by the Logos Model however was also influenced by another characteristic of the psychological model, namely, the fact that the human mind makes no use of algorithms as it "understands" a sentence. The mind can be said rather to function opportunistically, by seizing what it needs in order to understand. This fact of the psychological model led us away from algorithmic formalisms in the direction of more purely heuristic procedures. This bias against algorithmic solutions was further strengthened by the realization that algorithms in fact work well only with formal or artificial languages and in fact cannot cope with natural language qua natural language. As I said, the spectre of logic saturation was one that haunted our earliest reflections on the processing of natural language in a machine environment, and that led us eventually in a direction away from the Turing Machine model, and by an extension of Church's thesis, from all algorithmic formalisms. But having said mis, neither is it true that the alternative to an algorithm could ever be merely a series of ad hoc procedures, as this is just another path to logic saturation, to chaos.

We are now at the point in our discussion where it is appropriate to introduce the new "idea" that shaped the evolution of the Logos Model perhaps more than anything else. The idea was that language itself rather than the machine (with its algorithm) should drive the process and also provide order. This relates of course to the current recognition in MT circles that the text should drive the process. But I believe this is carried out in a

more radical way in the Logos Model, such that the sentence itself now literally becomes the algorithm that shapes and controls the analysis, indeed that controls everything that happens to it, an extremely efficient algorithm moreover that has the critical property of being resistant to complexity. Let me explain how this works.

It has to do with the nature of the Logos rule base, with the fact that the rule base is written in SAL and that this is the same SAL language in which the input sentence is represented from the very outset of analysis. This means that the elements of the SAL string now can serve as search arguments to the rule base, much as words serve as search arguments in dictionary lookup. This homogeneity between rule base and input stream has a number of positive implications. For one, it means that the rule base can grow arbitrarily large, much as a dictionary can grow large, with strictly sublinear impact on the search time. It also means that a new rule can be entered into the rule base with a high degree of assurance that it will be looked at if it should be looked at, and not looked at if it should not be looked at. Finally, it allows for the very interesting situation where the rules are self-organizing, much as a dictionary is self-organizing. This latter feature has profound implications for the problem of "conceptual complexity" and the ability of the mind to keep track of the rule base, what it covers and what it does not cover, and so on. Many years ago David Hayes said that one of the most basic problems in machine translation was the question of the order of the application of rules. The new "idea" embodied in the Logos Model entails a complete solution to this problem .

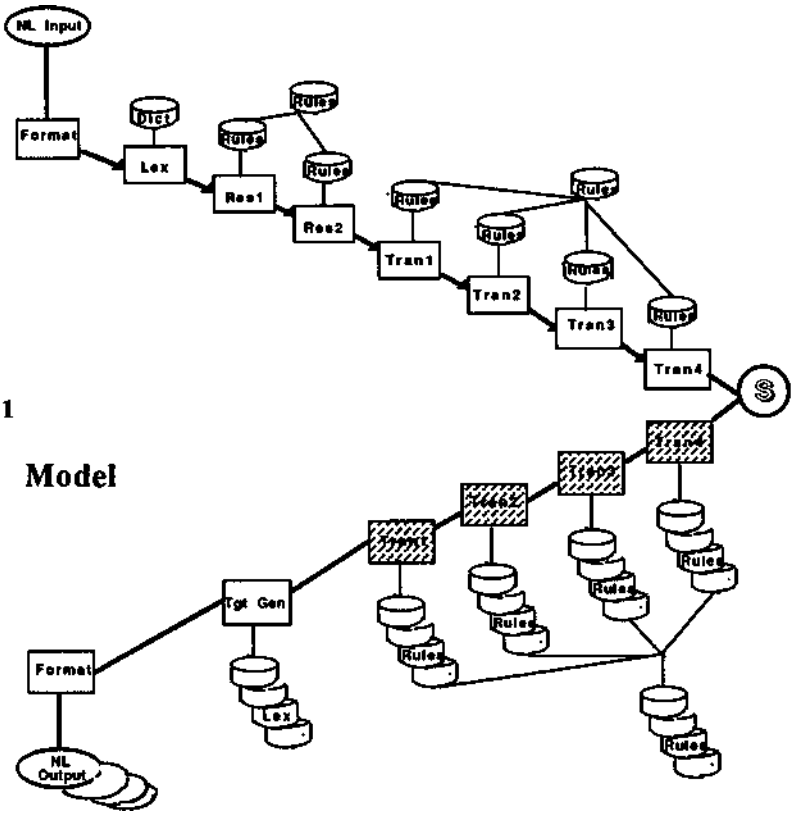


Figure 1

The Logos Model

The System Flow

As an input sentence enters the pipeline, it immediately undergoes conversion to an SAL string. This SAL string then passes through a series of "deterministic" parses, to rid the string of ambiguity and to simplify it. As the SAL string passes down the pipeline, the string itself is dictating what SAL rules should be applied to it (recall that the rules and the string look alike). But the rules also interact on the string, incrementally giving it more precise semantico-syntactic definition and greater simplicity (by reducing the string to its heads).

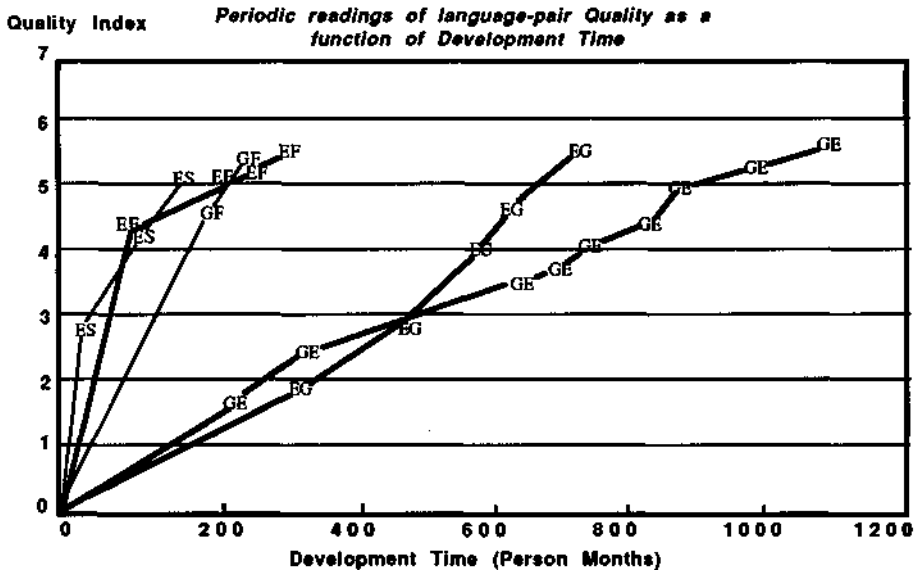
SAL is a semantico-syntactic language. The integration of semantics and syntax at virtually every step in

this process accounts for the determinism of the process. Only one path is ever effected, and this is done without backtracking. (A judicious use of "look-ahead", to detect "garden-path" situations for example, makes backtracking unnecessary.) SAL rules can reverse some previous analysis, but this is not usual. More typical is the tendency in a given analysis to only partial resolution, leaving it to a subsequent stage to complete the job. For example, an SAL rule in RES2 will see that "das Maedchen" is singular and either nominative or accusative. Not until as late as TRAN4 will an SAL rule decide which case applies. Or again, RES2 will resolve the "das" in "das Maedchen" to a determiner, but in the expression "das Bier" the resolution of the "das" is deferred to TRAN2 where other possibilities are considered (e.g. "Das Maedchen, das Bier bringt,..."). The general rule of analysis is to resolve issues "as early as possible and as late as necessary". This indicates that linguists have a great deal of freedom in deciding where to do things in the pipeline. The software facilities available are essentially the same from module to module, but the use made of these facilities is often quite different in the German and English sources.

Although the SAL rule base differentiates into three distinct classes of rules, lor source analysis, transfer, and generation, and although these rules stand as separate and distinct rule bases, the execution of these rules are in fact normally interleaved in a manner specified by the linguist. Thus, although transfer at the macro level normally occurs only after analysis has been completed, transfer of micro elements will have normally taken place before that analysis has been brought to a completion. The principle in all this is to do things at the most logical point, at the point where the linguist (source or target) knows that what he is doing is correct.

Perhaps the most unusual fact about the Logos Model is the great number of rules distributed across the six principal stages of analysis (RES & TRAN). In the English source system, for example, there are well over 10,000 rules altogether. This can be construed as either a sign of weakness or a sign of strength. It is certainly a sign of the system's semantico-syntactic comprehensiveness and of the fact that the system deals with a great quantity of linguistic phenomena not easily handled at the lexical level, or that is being handled more generically and therefore more efficiently. The rules are for the most part shallow rules, encoding myriads of mostly generic linguistic facts which when present in an input string have a very good probability of being handled correctly. The SAL rule base can be thought of as a dictionary of semantico-syntactic patterns, indexable like a dictionary, and like a dictionary, indefinitely extendable.

Fig. 2



Evaluation

The Logos Model exhibits unusual capacity for improvement. It is an open-ended system with seemingly unlimited capacity to absorb new linguistic data and make proper use of it. The data in *Fig. 2* is indicative of this. (The Quality Index readings in the chart are produced by an outside source under contract to the Company.) There are two interesting curves in this chart warranting comment. One is the relative speed with which a new target language can be brought to 90% of optimum (within 12-18 months). The second is the steepness of the Quality curve, indicating that the Logos Model has not yet reached its full potential.