

The evolution of ideas in classical MT : B. Vauquois' contribution to the theory and practice of MT (1960—1985)

Christian BOITET
GETA, Institut IMAG
(UJF & CNRS)
BP 53X, 38041 Grenoble Cedex, France

Abstract

B. Vauquois has been one of the foremost researchers in MT from 1960 until his death in October, 1985. His contribution to the field is presented, as well as the evolution of his ideas, which have influenced the design and construction of several MT systems, in France and abroad. The paper tries also to generalize and offer some general insight into some known traps and illusions encountered when building MT systems.

Introduction

Machine Translation has been fifty years old in 1989. First generation systems have long been commercially available (Hutchins 1986), while some mature systems of the second generation begin to appear on the market. After 15 years of purgatory following the (in)famous ALPAC report of 1966–67, MT has come to be fashionable again, and many projects have been and are started. In other scientific branches, people try to capitalize on previous results. Mathematicians study and redemonstrate theorems, physicists reproduce experiments, computer scientists read "The Art..." and dissect compilers. In MT and some other areas in computational linguistics, however, research seems to function "as a finite-state device" (a comparison made by J. Rouault's at a Grenoble meeting on man-machine communication in 1988), that is, with bounded memory.

For two years, I have reviewed all papers authored by B. Vauquois, in order to prepare a selection of them for a book (published in 1988–89), and have been struck by the truth of the preceding remark. Many things which were discovered and applied in the sixties are just being rediscovered now, and used in "new frameworks". But the absence of memory leads to the same illusions, the same traps and the same backtracks as those of the past, and probably to similar successes and similar failures. After all, language has not changed so much since, and pioneers are often more innovative and sharp than their followers.

Hence, it may be useful to retrace B. Vauquois's contribution to MT, and the evolution of his ideas. For more than 25 years, he has been active as a researcher, a teacher and an implementer in mathematics, computer science and machine translation. His initial training was in mathematics and astronomy. He became interested in computer science while working on his main thesis (on astronomy), around 1955, when the field began to exist (in France). For his second thesis, he was asked to explore obscure questions of decidability. This is how he was introduced to the fundamental domains of logic and formal languages.

Enthusiastic about these new horizons, he accepted without hesitation Pr. J. Kuntzman's proposal to come to Grenoble and launch a curriculum in computer science and formal languages, as well as a research laboratory on MT. This was in 1961. CETA (Centre d'Etudes pour la Traduction Automatique) had been created recently, under M. Sestier's direction, with the participation of some young collaborators whose names are well known now (J. Pitrat, M. Gross,...) : it was reorganized as CETA-P, in Paris, and CETA-G, in Grenoble. Some time afterward, M. Sestier decided that the obstacles faced by MT were too fundamental, and without doubt insurmountable. He resigned, leading to the collapse of CETA-P. Having already seen some new angles of attack, B. Vauquois continued with the now unique CETA, a "proper" laboratory of CNRS, which created the first and only complete second generation MT system of the sixties.

This first episode is instructive. Some eminent researchers have persisted in claiming that MT was impossible in principle, even in the face of evidence to the contrary provided by real-size laboratory prototypes as well as by operational systems such as METEO (which translates more than 100 standard pages a day, with fewer than 5% corrections, every day since more than ten years). At the beginning of aviation, it had analogously been stated, and "demonstrated" with deep philosophico-scientific arguments, that planes could never fly. But planes do fly, and MT systems do translate. However, planes do not swing their wings, and MT systems do not translate as humans do.

As a matter of fact, there is no reason why MT systems should imitate the human behaviour of translators, and give identical results.

The only important thing is that their output be a kind of translation acceptable to humans for some tasks such as information-gathering or professional post-editing.

Also, a common illusion is that human translators produce perfect, or "high quality" translations. But, if this were true, there would be no need for revisors in the first place.

Of course, unrealistic goals of the beginnings have been replaced by more pragmatic ones, in all places where real MT systems have been successfully built. Even Bar-Hillel, a vocal critic of the sixties, revised his opinion (as recorded in the feasibility study of MT conducted at Austin, Texas, in 1971 [51]). In France, the persistence of B. Vauquois, astronomer-mathematician turned computer-linguist, and the talent of the team he had gathered, motivated and guided, had already demonstrated *ad absurdum* the falsity of the initial pessimistic predictions. By the end of the sixties, the Russian-French translations obtained and evaluated on a very important volume (more than 400,000 running words, or 1,600 standard pages, that is, quite more than for later university systems) were of such quality that, when I first saw them in October 1970, I was tempted to look for another research area.

1. B. Vauquois as a researcher : the ideas behind MT systems

B. Vauquois' contribution to MT is prominent because of the length of his work and the variety of his competence. Unlike many others, he did not change his area of research every fifth year or so, but concentrated on MT (renamed MAT around 1980, see below) and more generally on NLP (Natural Language Processing). But he introduced at least one important concept every fifth year. Also, his contribution has always been at the frontier between linguistics, computer science and mathematics, on which the field of NLP is based.

1.1. 1960–65 : the enlightened application of the theory of formal languages

The first thing B. Vauquois did when he started to work in this new domain was to study all that had been done before, in the US and in the USSR, since the heroic postwar beginnings (around 1949). Heroism is the proper term, considering that researchers of that time had access only to a few K of memory, and to machine language (macroassembler became available only several years later).

The "first generation" systems created by the first pioneers did not incorporate any method from formal linguistics, such as the use of regular languages or syntagmatic grammars, for the good reason that these theories were formulated some ten years after research on MT begun. These systems contain neither tree structures nor grammar rules, but programs which work on a linear representation of the text, where a fixed-size area, containing directly coded word-specific information (e.g., bit 3 in byte 35 for "singular") as well as binary relations between words (expressed as pointers contained in some subareas), is associated with each occurrence.

B. Vauquois also became acquainted with the new trends in formal linguistics and formal languages, initiated by Chomsky and Schutzenberger in the US, Mel'tchuk and Gladkij in the USSR (the reader should pardon this somewhat exaggerated reduction!). Very early, he became a member of the committee on ALGOL-60, the first programming language for which syntax has been defined by means of a formal grammar. With F. Génuys and J. Poyen, he published an annotated translation of the famous "report on ALGOL-60" in Chiffres (1960, n° 3). This aspect of his work has certainly influenced CETA's research.

As a matter of fact, B. Vauquois resolutely led CETA in this new direction, by proposing to use the formalism of context-free rules in Chomsky's normal form to describe the syntax of the languages to be analyzed (Russian, German, and some Japanese). Recall that rules are of the form $A \rightarrow BC$, or $A \rightarrow t$, where A, B, C are non-terminals (names of syntagms, as "nominal phrase"), and t a terminal (a word, or, more often, a morphosyntactic class such as "verb", "adjective"...). This choice was dictated by considerations of algorithmic efficiency : with this type of grammar, it is possible to use Cocke's algorithm (later called Cocke-Kasami-Younger algorithm), which is much more efficient than those associated with other normal forms such as Greibach's. It uses dynamic programming : the analyzer obtained is bottom-up, multiple ("all-path"), and factorizing ; its worst time is $O(n^3)$ for a string of length n .

The grammar formalism and the algorithm were improved very quickly (Veillon 1970). As soon as 1963–65, some optimizations lowered the complexity to $O(n^2)$ for most grammars, attributes were added to the basic symbols (terminals and non-terminals) to represent grammatical properties (gender, number, etc.), and conditions/actions were added to the rules, in order to test certain congruences on the right hand sides (agreements in case–number–gender, for instance), and to synthesize the attributes of the left hand sides. A certain control on the derivations was also introduced, by introducing validations and saturations (invalidations) between rules, in order to reduce drastically the number of structural ambiguities. This is a constant guiding principle of B. Vauquois' approach :

always try to extract the best from methods used in general computer science, and apply it to NLP, even if that leads somewhat away from the current (formal) linguistic orthodoxy.

Note that the use of complex attribute or feature structures on tree nodes began to be popular in formal linguistics only 15 years later, in the "XYZG" formalisms.

At the beginning of MT, an analogy was made between translation and decoding (W. Weaver, 1949), but its absence of pertinence was soon realized. Probably, credit must be given to CETA to have introduced, on B. Vauquois' initiative, the far more fruitful idea that

MT systems and compilers have much in common.

As a matter of fact, because, in first generation as well as in second generation systems (1G and 2G, in the following), no explicit representation (in the computer) of the domain of interpretation of the texts to be translated is available, the task is reduced to transforming the form while preserving the content. In the same way, a compiler transforms a program into an equivalent program, that is a program which computes the same function, without recognizing that function as such (this is in fact a well known undecidable problem).

Hence, a 2G MT system is seen as a sort of "natural language compiler". This analogy inspired and justified the construction of computer environments for linguistic programming specialized to "generate" MT systems, quite long before the emergence of "compiler compilers" and "software engineering environments" for programming languages. In the technical history of MT, this kind of dialectics between linguistics and computer science has happened more than once, each in turn borrowing from the other tools or concepts to be used in this ambivalent application.

1.2. 1965–70 : the integration of modern linguistic theories

The analogy with compilation also supports the proposal made by Y. Yngve, in the sixties, to organize an MT system in three logical steps, a monolingual analysis, a bilingual transfer, and a monolingual generation. Then, the task of the analyzer is to produce, for each unit of translation (a sentence, at that time), a "structural descriptor" containing no reference to the target language. But... what kind of descriptor, exactly ?

Around 1963–64, first versions of the Cocke based analyzers were ready. But the descriptors obtained appeared too specific (to the language *and* to the grammar). For reducing the work of transfer as much as possible, it was evident that a description as universal as possible was needed. This is the basic idea of the "pivot language". B. Vauquois looked for the most promising linguistic theories, and selected those by Tesnière (the "actants") and by Mel'chuk (the "meaning \leftrightarrow text" model). Finally, he proposed a "hybrid pivot" (a term coined by Shaumjan), where the grammatical and relational symbols are universal (abstract time, semantic features, actancial or circumstantial relations such as "agent" or "cause", "consequence", ...), while the lexical symbols are taken from a natural language. In an MT system, transfer is then reduced to a "lexical" transfer, realized by means of a bilingual dictionary which enables the determination of the best equivalent of a term as a function of its context in the structure produced by the analyzer.

The nature of the "interface structures" representing units of translations is crucial in the design of an MT system.

As far as this "hybrid pivot" approach is concerned, we will see below why B. Vauquois abandoned it later. A complete discussion of the merits of the various "pivot" and "transfer" approaches is outside the scope of this paper. Suffice it to say here that a pivot language with universal, language-independent lexical symbols seems to be quite better, because it reduces the number of lexical transfers to $2n$, in a multilingual system with n languages (and thus at most $n(n-1)$ language pairs). However, a universal lexicon of language-independent concepts is quite difficult and costly to construct.

The TITUS system (of the Institut Textile de France, see Ducrot 1973, 1982) has been the only MT system to employ this "pure pivot" technique until 1980, in a very specific setting (severely controlled, ambiguity-free input language and domain/typology restricted to textile abstracts). After 1980, Fujitsu and NEC have also selected this design, for MT systems of a far more general nature. Other solutions involve other types of "impure pivots", such as Vauquois' "m-structures" – see below –, without a universal vocabulary of concepts, but still with a linear number of lexical transfers, actually, at best, $2n-2$ for n languages.

1.3. 1970–75 : modular grammars and heuristic methods

In 1969–70, CETA tried to sum up its first experiment, and to evaluate the Russian-French system, mainly because the IBM 7044 on which the system had been developed was to be replaced by an IBM 360/67 under CP/CMS, the first computer equipped with virtual memory and the first O/S implementing the concept of virtual machines. After this first conclusion, CETA gave birth to three smaller units. One, under J. Rouault, turned to the study of some theoretical problems in linguistics, and later to their application in documentation systems. A second was founded by G. Veillon and switched to the new field of AI, progressively abandoning NLP for robotics and now vision. The third, GETA, representing roughly half of the previous team, continued to work on MT, under B. Vauquois' direction.

The objectives as well as the methods of MT were reviewed in detail [26, 30]. GETA tried to capitalize on its recent experience as well as on more recent attempts by other groups, most notably TAUM (Montréal). It was already clear for the specialists that a *general* Fully Automatic High Quality Machine Translation (FAHQMT) system was a utopia : this would be to ask more than from the best human translators. In professional contexts, raw translations are always revised (for one standard page of technical text, the average times are 20 mn for revision and 1 h for translation). The possible goals of MT were made clearer: .

First, MT systems should be designed to produce useful and cost-effective translations.

Second, they should be organized to facilitate construction, debugging, maintenance and evolution.

Third, they should give the possibility of implementing a variety of linguistic frameworks.

In other words, MT outputs should be either *raw translations* of a quality comparable with that of human first draft translations, that is subjectively acceptable by the revisors and revisable with comparable effort (thanks to good grammaticality, and to the presence of marks for alternate lexical choices, and of warnings in case of doubt on the construction, or on some grammatical property or function,...), or *rough translations*, enabling the reader to understand the meaning of the original document, without revision, and by speedreading (there should be as few marks or extratextual symbols as possible, and grammaticality is less important).

Office automation techniques began to be introduced around that time, and opened a whole range of new possibilities for automating translation, ranging from MT of the old form, not meant to be revised and essentially useful in information-gathering contexts (e.g., access to scientific and technological textual databases), to CAT (Computer Aided Translation), where useful computer tools are integrated on a workstation for human translation or revision. This is of course reminiscent of CAD, CAM, and the like. The whole set of techniques has been renamed M(a)T or MAT, for Machine (aided) Translation, to retain the "MT" flavor. "MT" itself now covers both UMT (Unrevised MT) and RMT (Revised MT), while IMT (Interactive MT, a concept first experimented on at BYU (in Provo, Utah) in the seventies) is still another intermediate possibility.

B. Vauquois choosed to concentrate on RMT, with the supplementary constraint of trying to translate a text completely, even if some utterance does not conform to the implemented linguistic model. This was an important shift from the previous approach, a lot more comfortable (for MT builders, not for users), because not completely analyzed sentences were rejected and translated word for word. But the experience gained with the tuning of a large grammar for Russian had once more illustrated that language is "productive", a constant claim of linguists.

Let us digress briefly here and consider the interesting possibility that this claim might be true in its mathematical sense. Then, any finite grammatical and lexical description D of a natural language L , generating an infinite (recursively enumerable) set T of "valid" utterances, would be constructively non-exhaustive, that is, a valid utterance U not in T could be obtained algorithmically as $f_L(D)$. This hypothesis, an analog to the Gödel-Rosser incompleteness theorem, is based on the intuitive analogy between judgements of truth (of formulæ) and of grammaticality (of utterances). Because of the lack of a formal characterization of the class of natural languages, it can not be demonstrated mathematically, and may be at best considered as a metamathematical "thesis" like Church's thesis. *If it is true, as practical work on natural language suggests, then there can be no exhaustive description of any natural (sub-)language.* In any event, it was no doubt wise of B. Vauquois to insist that

MT systems should always be equipped with fail-safe devices.

This shift in the definition of the goals of MT had an important consequence on the method. B. Vauquois decided *not to base the production of structural descriptors on analysis models, but on transduction models*, designed to produce an output on any input.

Having observed that large attributed grammars of several hundreds of rules became too unwieldy, he asked for a more modular organization. Last, but not least, he recognized the problems of a totally combinatorial approach, and looked for some means to implement heuristics in linguistic programming.

J. Chauché worked on these three ideas, and presented a solution in his 1974 doctoral thesis [72,73], in the form of two specialized languages, ATEF ("Analyse de Textes en Etats Finis", based on a finite-state, non-deterministic model) for morphological analysis, and CETA ("Contrôle Et Transduction d'Arborescences", based on tree transduction using rewriting rules), the ancestor of the current ROBRA [66], for structural analysis, transfer and generation. In ATEF, each morphological "format" (template) associated with a "morph" is also the name of a subgrammar, called when the corresponding morph is segmented. In ROBRA, a transformational system consists in a "control graph" with conditions attached to the arcs and transformational grammars to the nodes. Each grammar is an ordered subset of the (unordered) pool of rules. This way, modularity is achieved, because the number of rules may be very large, while each grammar contains in general a small number of rules, executed in parallel on the current "object tree". The non-determinism of the traversal of the control graph (unary backtrack), together with the possibility to recursively call subgrammars or subsystems on subtrees, give the means for heuristic programming, more or less guided by the data, according to the strategy chosen by the linguist.

1.4. 1975–80 : the "multilevel" structural descriptors

As soon as the new computer tools were available, B. Vauquois tried to improve the linguistic organization of the translation process. The preceding experiment had shown that a sequential organization, computing level after level of linguistic interpretation, posed numerous problems. For instance, the selection of a syntagmatic structure could not use information from the upper level of semantic relations, which was computed later. In many cases such as

"He types on a computer on his desk on weekdays",

however, this "interaction" between levels is useful, if not necessary. In order not to choose too early between competing analyses, then, a sequential organization should allow some backtracking, or keep all possibilities until the end of analysis.

Of course, it would be possible to retain only the invariant upper levels of interpretation after analysis (logical and semantic relations, semantic features, abstract determination, actualisation, quantification, etc.). But, as the previous experiments had shown, translations obtained by this method cannot be really translations, because it is impossible to produce the desired parallelism in style, as all trace of the surface expression is erased. For example, it is not possible to translate the English passive by the French reflexive, in appropriate contexts (*many equations are solved by iteration* —> *beaucoup d'équations se résolvent par itération*). Also, there existed (and exists) no satisfactory universal description of the time / modality / aspect. For such facets of languages, translation necessitates some contrastive, language-dependent treatment.

Another severe problem with a "grammatical pivot" approach is the "all-or-nothing" problem : no translation is possible if analysis has not produced a correct result in terms of semantic relationships, the most difficult task. If the size of the unit of translation becomes larger than one sentence, e.g. one or several paragraphs, which is desirable for a number of reasons (resolution of anaphoras, actualisation agreements, theme/rheme coherence, in particular), it is almost certain that the result of analysis will not be complete, and hence that the majority of units will not be translated.

Independently of what was done about the same time in the US, in the framework of a very important ARPA project on speech recognition, B. Vauquois invented a sort of "blackboard" technique for MT. The idea, first exposed at a meeting of the Leibniz group in 1974, is to associate with a text (the units of translation were not reduced to sentences any more) a unique structural descriptor, in the form of a tree decorated by attributes corresponding to the various levels of interpretation, in such a way that the geometry of the tree be close to that of usual syntagmatic or dependency trees, and that it be possible, by means of simple transformations, to produce from it the structural descriptors of the logical and semantic levels.

Interface structures should provide several levels of interpretation.

These "multilevel" structures, later also called "m-structures" by analogy with the c-structures and f-structures (constituent and functional structures), offer a means not only to let levels "interact", but also to tolerate errors and implement fail-safe strategies. If an utterance cannot be analyzed completely at the highest level of interpretation, it is still possible to translate the faulty part (and not the whole utterance) by starting from the next lower level, which acts as a kind of "safety net", and not word for word. On the other hand, m-structures are also *multiple* structures, in the sense that they are used to encode unsolved ambiguities, which makes it possible to treat ambiguities *directly*, and also to avoid the combinatorial explosion on long utterances : until now, this is the only technique enabling an MT system to translate not sentence by sentence, but paragraph by paragraph, or even page by page.

1.5. 1980–85 : the "static grammars", a specification language

These ideas were soon used in practice. In 1980, GETA had developed a new, complete Russian–French system, and the kernel of an analyzer for German. In cooperation with SFB/100 (Saarbrücken) and UNICAMP (Campinas), several versions of analysis of French and a large analyzer of Portuguese, with mockup transfer and generation into English, had been implemented. During the summer of 1981, a first English–Malay prototype was implemented at USM (Penang). Around 1981, a feasibility study on French–English began. B. Vauquois participated very actively in the work on the last two systems, designing and implementing large parts of their grammars.

He then realized fully the problems of lingware engineering posed by the writing of large transformational systems. For the linguistic computation itself, he had almost completely abandoned the strictly declarative formalisms, inevitably leading to a fully combinatorial approach. He felt necessary to return to declarativeness at a higher level, that of specification.

Way back in 1980, I remember that he was trying to represent the correspondence between utterances and associated m-structures by means of simple diagrams [43, 45–48]. No existing formalism was adequate. As a matter of fact, all supposed a context-free base, and projective correspondences. On the other hand, all were asymmetric, giving more importance to the strings than to their structural descriptors. Last, none was adapted to the m-structures. This is why B. Vauquois and S. Chappuy, who wrote her doctoral dissertation on the subject, invented a new, bidimensional formalism, which they called the "static grammar" formalism. Each rule is a "board" ("planche" in French) containing various elements, one of which is a diagram expressing the geometry of the partial correspondence described by the board.

The term of "static grammar" is somewhat infelicitous. It has been coined in opposition to that of "dynamic grammars" used to speak of the rule systems implemented (in ATEF, ROBRA, etc.) for analysis, transfer and generation. The term of SCSG (static correspondence specification grammar) has since been proposed. Hundreds of boards have been written, for several languages, and in particular for French and English, in the framework of the MAT–NP (French MAT National Project, 1983–87), and used for writing analyzers and generators. This is an example of a clearly useful formalism, which has quite clear "intuitive" semantics, but for which adequate formal semantics have been difficult to formulate (this has entailed modifications of the initial formalism), perhaps because of the original practice-oriented motivation. This is a part of Zaharin's contribution in his thesis [129]. Based on this modified model, an interpreter is now being implemented.

Let us give a flavor of SCSG by using an example involving the strictly context-sensitive formal language $L = \{a^n b^n c^n\}$, which can be used to model natural language constructs of the "respectively" type, such as :

<i>Linguists,</i>	<i>lexicographers</i>	<i>and engineers</i>
<i>describe,</i>	<i>index</i>	<i>and implement</i>
<i>grammars,</i>	<i>dictionaries</i>	<i>and NLP-systems.</i>

At least two structural descriptors seem "natural" for the string $w_n = a^n b^n c^n$, namely

$$(T1) \ S(A(a_1, A(a_2, \dots A(a_n) \dots)), B(b_1, B(b_2, \dots B(b_n) \dots)), C(c_1, C(c_2, \dots C(c_n) \dots)))$$

$$(T2) \ S(a_1, b_1, c_1, S(a_2, b_2, c_2, \dots S(a_n, b_n, c_n) \dots))$$

Tree T1 gives rise to a projective correspondence, but does not imply any particular relation between a_i , b_i , and c_i . If we want to consider the triples a_i , b_i , c_i directly as discontinuous constituents, T2 is better. T1 can obviously be produced by attaching adequate translations to the rules of a right linear grammar (such as $S \rightarrow ABC$, $A \rightarrow aA / a$, $B \rightarrow bB / b$, $C \rightarrow cC / c$), but such a grammar cannot characterize L . It seems that T2 cannot be obtained in an analogous way, even taking a context-sensitive grammar for L as a basis. But there is a very simple SCSG of just two rules which describes the *correspondence* between L and its T2 descriptors.

$$(R1) \ [S(a, b, c), abc] \longrightarrow [a, a] [b, b] [c, c]$$

and

$$(R2) \ [S(a, b, c, S(\$F)), \alpha\beta\gamma] \longrightarrow [a, a] [b, b] [c, c] [S(\$F), \alpha\beta\gamma]$$

with $\alpha \in a^*$, $\beta \in b^*$, and $\gamma \in c^*$.

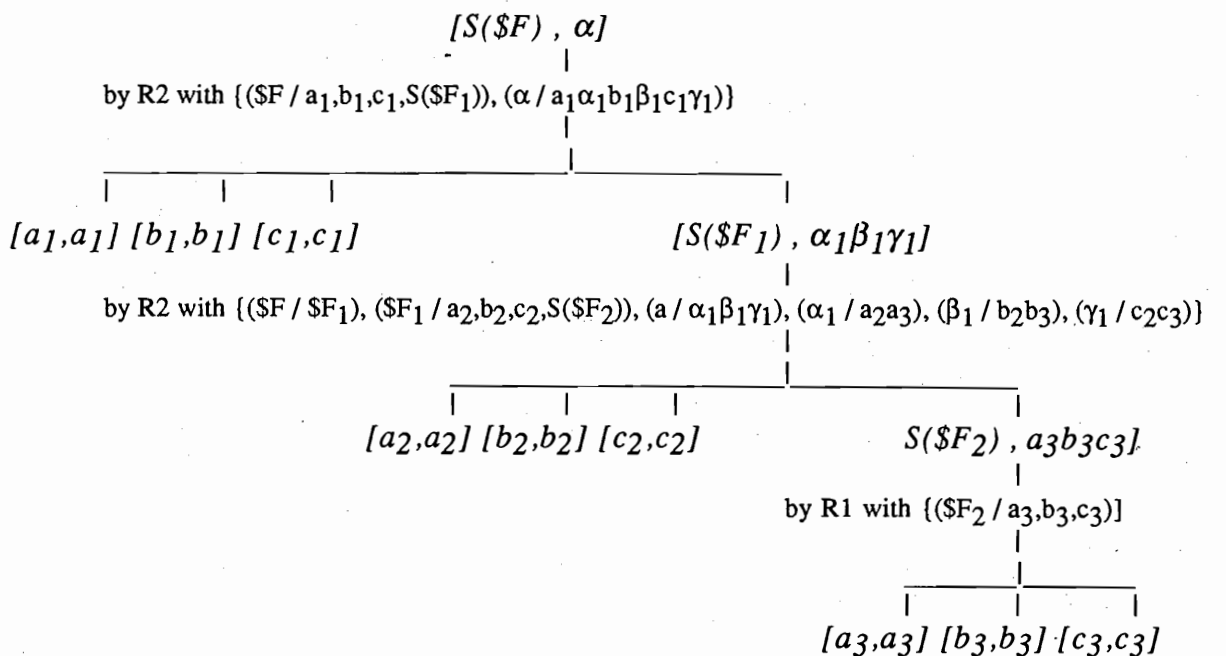
$\$F$ is a forest variable, and α, β, γ are string variables. The axiom is $[S(\$F), \alpha]$, and the terminals are all the pairs $[x, x]$ made of a one-node tree labeled by the letter x and of the string x . The generated correspondence language is the set of all variable-free pairs $[tree, string]$ derivable from the axiom in the usual fashion, with adequate substitutions of variables at each step.

The *with* condition ensures that no parasite correspondence is generated.

For example, the correspondence

$$[S(a_1, b_1, c_1, S(a_2, b_2, c_2, S(a_3, b_3, c_3))), a_1 a_2 a_3 b_1 b_2 b_3 c_1 c_2 c_3]$$

can be derived as shown in the following derivation tree :



Here, we have used indices to make the subcorrespondences clear. It is easy to augment slightly the formalism to describe them within the rules, by indicating on each node to which substring it corresponds, and to which substring the subtree rooted at it corresponds, the substrings being possibly discontinuous, as in the previous example [69].

1.6. 1985— : towards a new treatment of ambiguity

In NLP, the omnipresent and fundamental problem of ambiguity is often maltreated, badly treated, or not treated.

The first technique consists, as soon as an ambiguity appears, in deciding for a solution on the spot, without possibility of ulterior correction. It is that of 1G MT systems. If backtracking is introduced, as in many Prolog-based analyzers, the *first* correct solution is obtained, but nothing guarantees that it is the *best*, or even a good one.

The second approach, known as the "filter" approach, is combinatorial : all possibilities are computed, and further more and more stringent well-formedness conditions eliminate spurious results. However, there remain in general several results, between which the choice is arbitrary.

A third idea is to compute a "valuation" on the results, by assigning to them a coefficient computed step by step during the construction process. Apart from the fact that such a function is difficult to define, several "better" results may also be obtained. However, for some reason not yet explained by linguistic theories, this works rather well in practice [115].

Still a fourth way is to use preference rules between two or more structure patterns, corresponding to the same substring. This idea was used in the CETA system, before 1970. Under a somewhat different form, it has been popularized in the seventies by Y. Wilks ("preference semantics"). This type of method is the only one to allow for a real treatment of ambiguity, represented as such.

The heuristic methods developed by B. Vauquois between 1970 and 1983 belong to this type. Ambiguities are treated directly, by means of an appropriate coding in the structure representing the unit of text processed. However, in a given state of this structure, on a given branch of the computation tree (recall that ROBRA's highest level of control consists in unary backtracking), only a subset of the ambiguities is represented. This technique gives quite good results, but its implementation may be delicate.

This is why, during the period of linguistic specification of the MAT-NP, from November 1983 to the end of 1984, B. Vauquois imagined to add to the "construct boards" of the "static grammars" "ambiguity boards" which are really preference rules between two structure patterns, for a given string pattern. In July 1986, after 18 months of development, the static grammar of French contained about 170 construct boards and 350 ambiguity boards. The corresponding "dynamic" structural analyzer was 18000 lines long (in ROBRA source code).

Immediately, B. Vauquois initiated a new research on the treatment of ambiguities (Y. Lepage), thinking that it would one day be possible to precompute a certain number of ambiguity cases from a SCSG, in function of certain constraints (on the length of the strings, the depth of certain embeddings, the contexts in the string and in the structure,...). Of course, not all cases could be precomputed, because this problem is already undecidable for the class of context-free grammars.

However, if such a research succeeds, it would among other things make it possible to determine in advance until what stage one should perform a factorizing computation (such as "chart parsing" in a "dynamic" analyzer) before solving certain types of ambiguity. It would also open the way to a new architecture of MT systems, inspired by that of expert systems, and made of a purely declarative part, consisting in the lexicon(s) and the construct boards, of "metarules" describing how to solve the precomputed ambiguities, and of a general heuristic control, governing the whole and programmed by using high level primitives [129,130].

2. B. Vauquois as an implementer : the methodology of building MT systems

By initial training, B. Vauquois was not an engineer. But he certainly became one in his activity as computer-linguist. Although the term "computational linguistics" is horrible in French ("linguistique computationnelle"), he liked to use it in contrast to that of "formal linguistics" ("linguistique formelle"), and he always insisted that

There is an imperative necessity not to limit oneself to purely descriptive formal linguistic theories, but to realize *computational* linguistic systems.

As a matter of fact, a recent book on "computational complexity and natural language" (Barton, Berwick & Ristad, MIT Press, 1987) seems to support this point, by showing that classical descriptive grammatical frameworks are computationally intractable. Hence, building computationally efficient NLP systems must involve some amount of procedural treatment based on knowledge of *the* particular grammar/dictionary constructed for the (sub)language at hand. In his last years, B. Vauquois strove in effect to achieve a synthesis between the precision of the mathematical formalisms, the richness of the linguistic descriptions, and the efficiency of the computations.

2.1. Erudition, eclectism and pragmatism in linguistics

B. Vauquois had read and assimilated all important works on the domain, from those of linguists such as Chomsky, Harris, Mel'chuk, Zhol'kovskij, Shaumjan, Tesnière, etc., to those of logicians and formal language theorists. His conclusion, still valid today, was that

No unique linguistic theory covers completely a significant part of the natural languages, lest even one : ideas of various origins can and should be adopted, adapted and integrated in NLP systems.

Scientific dogmatism was completely alien to B. Vauquois. In his quest to build running systems, and not intractable abstractions, he was also a pragmatist.

If certain elegant and pertinent concepts happen not to be reliably and efficiently computable in the grammars, or if they cannot be introduced in the dictionaries for reasons of cost or of intrinsic difficulty of indexing, they must be abandoned.

For instance, the organization of semantic features must be kept relatively simple, otherwise the dictionaries are guaranteed to be inconsistent. Also, the grammar writers cannot make the best use of them, and adaptation of a system to a new domain becomes quite difficult [96].

It has been claimed that it was premature to work on MT before having performed an exhaustive study of the considered languages. To this day, no exhaustive study exists for any language, and, if our hypothesis on the productivity of natural languages is true, none can exist. However, there exist many NLP and MT systems which actually run and are useful. B. Vauquois has shown us how an in-depth study of the actual problems, and not of imaginary ones, can lead to solutions which are effective and computable at a reasonable cost.

Let us give still another example of this pragmatic orientation. As it became clear that a "pure pivot", in spite of its elegance, was not going to give the best results, B. Vauquois turned to the transfer approach. But he showed that this apparent retreat could in fact lead to some progress. We have already mentioned the "safety net" view of the various levels encoded in the m-structures, and the possibility to control the parallelism (or difference) in style if necessary. B. Vauquois' idea was to generate the corresponding structures *not* during transfer, but during generation. A new, powerful idea emerged, namely to

Consider a target interface structure input to the generator as a complex object representing a family of possible paraphrases, coding some marks of doubt as well as some unsolvable structural ambiguities and polysemies, and containing "advices" and "orders" to the generator.

This way, structural transfers can be kept small, and the tasks of generation are properly handled only in the generator, and not duplicated in the various transfers into the considered language.

2.2. Adequate computer "metalanguages"

B. Vauquois has contributed much [13, 24, 33, 45—47] to the now popular view that

Linguists and lexicographers should work directly on the computer, by using adequate "metalanguages", or "specialized languages".

This is the only way for linguists and lexicographers to use their competences at best, without being burdened by ancillary tasks, such as implementing specific data and control structures, or having to resort to "slave" programmers to translate their wishes into working programs, with the constant result that their desires, imprecisely formulated, are also incorrectly implemented.

On the other hand, B. Vauquois refused the temptation of using universal metalanguages such as the Q-systems [75], after having tested them. His fundamental argument was that this leads to perform the simpler parts, like the morphological and lexical phases, in an unduly complicated and costly manner.

There is no panacea for linguistic programming. MT systems should offer several specialized languages, based on subrecursive abstract models of known complexity.

The requirement that specialized languages be *based* on subrecursive models is important, and seems quite abstract, in view of B. Vauquois' alleged pragmatism. But, apart of the fact that he was also an excellent mathematician, this requirement comes from some bitter experience with the first CETA system. Recall that there was a transformational component, used after the context-free analysis and the obtention of dependency structures. Sometime, the computer would run for hours at night, without producing anything! It must also be remembered that dynamic grammars, by nature incomplete and approximate, are modified very frequently, contrary to the grammars of programming languages used by compilers or even to classical management software.

However, theory and practice show that certain functions can be considerably easier to describe and compute by using a universal (general recursive) formalism with the power of a Turing machine. The solution proposed by GETA has been to implement extensions which give that power, thereby designing the syntax of the metalanguage in such a way that the sources of undecidability (infinite loops) can be detected statically, at compile time, because they correspond to the use of some "switches" ("free" iteration mode for ROBRA grammars) or to the inobservance of certain simple conditions (form of recursive calls in ROBRA, and absence of cycles in the various dependency graphs of rules and grammars in ATEF, ROBRA and SYGMOR). Such ideas could perhaps be also useful in the design of new general purpose programming languages.

2.3. The emergence of lingware engineering

As soon as March 1977, B. Vauquois organized in Grenoble an University–Industry seminar on the industrial perspectives of 2G MT. Since that time, he tried to evaluate the effort necessary to build a cost-effective MT system, under various hypotheses. He extracted the three main parameters to consider (language, typology, domain). During the following years, ongoing work on various mockups, several feasibility studies, and the development of a "preoperational" Russian-French translation unit [64] enabled him to estimate the direct and marginal costs, and to define precisely the "profiles" of the various members of an ideal MT development team.

It is during the ESOPE project of ADI, aimed at preparing the MAT-NP, that he realized his most complete study [39], the estimations of which were later quite well verified during the real development. With other members of GETA, he also prepared an important document on a "methodology of work for a linguistic team" [41]. Finally, in the same conditions, he prepared of complete one year training plan for newcomers.

We see here the human and most important part, of the lingware engineering which began to be outlined. The other more technical part is also very important. It concerns the methodology for writing "dynamic grammars" (study of corpus, specification in SCSG, definition of the general strategy, division in "modules", construction of the control graph, then of the grammars, and last of the rules) and dictionaries. From 1970 to 1983, dictionary problems had been given less importance than before, the emphasis being on grammars and strategies. But,

While the grammars of an operational, large-size MT system define its asymptotic quality, its main cost and average working quality come from its dictionaries.

For the development of dictionaries, B. Vauquois and its team developed an indexing method using "forms", analogous to the one independently developed at the same moment by the MU-project in Kyoto, and different from that of "indexing charts" previously used at GETA, but difficult to use by non-specialist terminologists. In the new approach, there is one "neutral" dictionary for each language, from which the MT analysis and generation dictionaries are automatically produced, thus avoiding to index a term twice. However, the transfer dictionaries remain directional.

Finally, we note here another import from computer science, coming from the fact that

Software and lingware development have much in common.

It is now clear that MT systems, if they are not extremely task-specific (as METEO or TITUS), are large or very large computer packages, and that, unless development methods of industrial type are applied to them, with all the necessary discipline and logistics, one will eternally remain at the stage of nonsignificative and useless mockups. Rapid prototyping is a good thing [97,98], but one should be cautious not to enter an infinite prototyping loop.

Conclusion

B. Vauquois was not only a researcher and and implementer, but also an extremely clear professor, and an active promoter of scientific international cooperation. One of the founders of the ICCL (International Committee on Computational Linguistics), he was also its president, and as such one of the main organizers of the COLING conferences, from 1965 until 1984.

He has left us before he could see the quality of the French-English system he had designed for the MAT-NP. But the pertinence of his concepts, the clarity of his teachings, and the efficiency of his method have enabled his team to pursue and complete the development of this system without him, within the allotted time, despite various non-technical obstacles.

No doubt B. Vauquois will be remembered as one of the most prominent figures of our century, in the domains of NLP and MAT. He has greatly contributed to the evolution of theory and practice of classical MT. His work will continue to inspire many researchers and developers, both in classical MT and other new developments.

Acknowledgements

I would like to thank to Ph. Chatelin for having asked me to prepare a first paper on this theme for a congress on the history of computer science in France held at Grenoble in May, 1988. Shortly thereafter, J. Carbonell, M. Tomita and S. Nirenburg kindly asked me to prepare an English version for the second conference on the theoretical and methodological issues of MT organized by the Center for Machine Translation at Carnegie Mellon University. That version turned out not to be a simple translation, but a complete reworking. The present paper is a slight reworking of it, prepared for the third Summer School on NLP organized by CNET, CERIL, ENSSAT and APAST at Lannion in June 1990 and then further improved for the ROCLing-III tutorial (September 1990). Thanks also to Gayle Satoh, for having helped me remove some of the grammatical "bugs" of the CMU version. I hope not to have introduced too many new ones in the adaptation process!

-o-o-o-o-o-o-o-o-o-

References

Bibliographical references of B. Vauquois

- [1] B. Vauquois (1958)
Première thèse: Etude de la composition lentement variable du rayonnement radio-électrique solaire. Deuxième thèse : Arithmétisation de la logique et théorie des machines. Thèse d'Etat, Paris, 1958, 62 p.
- [2] B. Vauquois (1960a)
Calcul vectoriel. Cours de TMP à la faculté de Grenoble, 1960, 40 p.
- [3] B. Vauquois, F. Génuys, J. Poyen (1960)
Rapport sur le langage algorithmique ALGOL 60. Traduction en français de : Report on the Algorithmic Language ALGOL 60, In : Chiffres, n° 3, 1960, 45 p.
- [4] B. Vauquois (1960b)
Allgemeine Methoden für Maschinenübersetzung. Kolloquium über Sprachen und Algorithmen. Berlin, 1960, 2 p.
- [5] B. Vauquois (1960c)
Le rôle des formalismes en traduction automatique. Application à l'étude morphologique du russe. Premier Congrès de l'Association Française de Calcul. Grenoble, septembre 1960, 8 p.
- [6] B. Vauquois, J. Veyrunes (1962)
Présentation de l'analyse morphologique du russe. Research report, CETA, G-100-C, février 1962, 26 p.
- [7] B. Vauquois (1962)
Langages artificiels, systèmes formels et traduction automatique. In : Information Storage and Retrieval. Volume 1, 001-026, 1963, et présenté à : NATO Advanced Study Institute on Automatic Translation of Languages, Venise, juillet 1962.
- [8] B. Vauquois, G. Veillon, J. Veyrunes (1964)
Application des grammaires formelles aux modèles linguistiques en traduction automatique. Colloque sur la linguistique algébrique et la traduction automatique, Prague, septembre 1964, et KYBERNETIKA, 3/1, 1965, 9 p.
- [9] B. Vauquois (1965)
Présentation du programme d'analyse morphologique russe. Research report CETA, G-152-A, février 1965, 24 p.
- [10] B. Vauquois, G. Veillon (1965)
Syntaxe et interprétation. Congrès AMTCL, New-York, mai 1965, 34 p.
- [11] B. Vauquois (1966)
Présentation du Centre d'Etudes pour la Traduction Automatique CETA du Centre National de la Recherche Scientifique. In : T.A. Informations n° 1, 1966, 1-18.
- [12] B. Vauquois (1967)
Le système de traduction automatique du CETA Communication présentée au Congrès d'Erevan, U.R.S.S., février 1967, 41 p.

- [13] B. Vauquois, G. Veillon, J. Veyrunes (1967)
Un métalanguage de grammaires transformationnelles. Applications aux problèmes de transfert et de génération syntaxiques. 2ème Conférence Internationale sur le Traitement Automatique des Langues. Grenoble, août 1967, et Research report CETA n° G-2300-A, 56 p.
- [14] B. Vauquois (1968a)
Structures profondes et traduction automatique. Le système du CETA In : Revue Roumaine de Linguistique, 13/2, 1968, 105–130.
- [15] B. Vauquois (1968b)
A survey of formal grammars and algorithms for recognition and transformation in mechanical Translation. IFIP Congress 68, Edinburgh, August 1968, 17 p.
- [16] B. Vauquois (1969)
Probabilités. Paris, Herman, 1969. Collection Méthodes. Edition revue et corrigée en 1978, 163 p.
- [17] B. Vauquois, G. Veillon, N. Nedobekine, C. Bourguignon
Une notation des textes hors des contraintes morphologiques et syntaxiques de l'expression. In : T.A. Informations, n° 1, 1970, & In : COLING-69, Stockholm, 1969, 27 p.
- [18] B. Vauquois (1970a)
Intermediate language for automatic translation. In : Natural Language Processing. Edited by Randal Rustin. Courant Computer Science Symposium 8, Dec. 20-21, 1971, New-York, Algorithmic Press, 1973, 191–192.
- [19] B. Vauquois (1970b)
Traduction automatique des langues. In : Techniques de l'Ingénieur, 1970, 5 p.
- [20] B. Vauquois (1971)
Modèles pour la traduction automatique. In : Mathématiques et Sciences Humaines, n° 34, 61–70, 1971.
- [21] B. Vauquois (1973)
Les systèmes informatiques et l'analyse de textes. Colloque de Strasbourg sur l'analyse des corpus linguistiques. Problèmes et méthodes de l'indexation maximale, mai 1973, 10 p.
- [22] B. Vauquois (1974)
Analyse, génération et transfert des langues naturelles. Table ronde CNRS, 12-13 mars 1974, 6 p.
- [23] B. Vauquois (1975a)
Some problems of optimization in multilingual automatic translation. First National Conference on the Application of Mathematical Models and Computers in Linguistics. Varna, May 1975, 7 p.
- [24] B. Vauquois (1975b)
Méthodes, résultats et bilan de la TA. Journées d'Etudes du Festival International du Son Haute-Fidélité. Stéréophonie, Paris, 1975, 17 p.
- [25] B. Vauquois (1975c)
La traduction automatique à Grenoble. Paris, Dunod, 1975. Document de Linguistique Quantitative n° 24, 180 p.
- [26] B. Vauquois (1976a)
Les procédés formels de représentation des structures profondes. Internationales Kolloquium, Automatische Lexicographie, Analyse und Übersetzung, Saarbrücken, 23-25/09/1976.
- [27] B. Vauquois (1976b)
Automatic translation. A survey of different approaches. COLING-76, Ottawa 1976, et In : S.M.I.L. Statistical Methods in Linguistics, edited by H. Karlgren, Stockholm, 1976, 127–135.
- [28] B. Vauquois (1977a)
Enseignement et utilisation des langues dans la Communauté. Introduction du Président de la Première Séance. 3ème congrès européen sur les systèmes et réseaux documentaires. Francir la barrière linguistique, Luxembourg, mai 1977, volume 2, 2 p.
- [29] B. Vauquois (1977b)
Calculabilité des langages. Cours de Logique et Programmation, Université de Grenoble, 225 p.
- [30] B. Vauquois (1977c)
L'évolution des logiciels et des modèles linguistiques pour la traduction automatique. Colloque Franco-Soviétique, Moscou, décembre 1977, et In : T.A. Informations, 1978, n° 1, 1–21.

- [31] B. Vauquois (1978)
Description de la structure intermédiaire. Réunion aux CE, Luxembourg, avril 1978, 27 p.
- [32] B. Vauquois (1979a)
The evolution of oriented software and formalized linguistic models for automatic translation or machine aided translation of natural languages. Mission en Chine, février 1979, 28 p.
- [33] B. Vauquois (1979b)
Aspects of mechanical translation in 1979. Conference for Japan IBM Scientific Program, July 1979, 52 p.
- [34] B. Vauquois (1980a)
La traduction automatique. In : Les Sciences du Langage en France au 20ème siècle. Articles recueillis par B. Pottier, Paris, SELAF, 1980, 799–824.
- [35] B. Vauquois (1980b)
Etude de la validité du formalisme choisi pour représenter la structure linguistique interface. Rapport de contrat CEE, avril 1980, 81 p.
- [36] B. Vauquois (1980c)
L'informatique au service de la traduction. Journées d'Etudes du Festival International du Son Haute-Fidélité, Stéréophonie, Paris, 1980, et In : META, Journal des Traducteurs, Montréal, 26/1, 9–17, mars 1981.
- [37] B. Vauquois, Chang M. S. (1981)
Documentation sur le modèle d'analyse morphologique de l'anglais. Version simplifiée pour les textes scientifiques et techniques. Research report, TELECOM, avril 1981, 26 p.
- [38] B. Vauquois (1981a)
Etude de typologie de textes. Ambiguïtés lexicales dans les textes de la revue Electronics. Research report, TELECOM, juillet 1981, 50 p.
- [39] B. Vauquois (1981b)
Traduction assistée par ordinateur. Formation de spécialistes. Préparation du transfert technologique. Projet ESOPÉ, Contrat ADI, mai 1981, 25 p.
- [40] B. Vauquois, P. Daun Fraga (1982)
Etude de faisabilité d'un système de traduction automatisée anglais-français. Research report, TELECOM, février 1982, 22 p.
- [41] B. Vauquois, Ch. Boitet, P. Daun Fraga, J.P. Guilbaud, N. Nedobejkine (1982)
Définition d'une méthode de travail d'équipe linguistique. Projet ESOPÉ, Contrat ADI, novembre 1982, 94 p.
- [42] B. Vauquois (1982)
Modelle für die automatische Übersetzung. Sonderdruck aus Automatische Sprachübersetzung, 192–208, Darmstadt, Wissenschaftliche Buchgesellschaft, 1982, aus dem Französischen übersetzt von Irmela Freigang.
- [43] B. Vauquois (1983)
Automatic computer aided translation and the arabic languages. First Arab School on Science and Technology. Applied Arabic Linguistics and Signal and Information Processing, Rabat, October 1983, 21 p.
- [44] B. Vauquois, Ch. Boitet (1984)
Etudes sur les travaux en TAO portant sur des langues asiatiques (japonais, chinois, malais, thaï) et étude des problèmes liés au traitement de textes utilisant de grands jeux de caractères (idéogrammes). R. R. n° 83/739, Ass. Champollion/ADI, juin 1984, 81 p.
- [45] B. Vauquois (1984)
The organization of an automated translation system for multilingual translations at GETA. IBM Europe Institute on Natural Language Processing, Davos, July 1984, 41 p.
- [46] B. Vauquois, Ch. Boitet (1985)
Automated translation at Grenoble University. In : Computational Linguistics, 11/1, 28–36, Jan.-March 1985.
- [47] B. Vauquois (1985)
The approach of GETA to automatic translation. Comparison with some other methods. International Symposium on Mechanical Translation. Riyadh, March 1985, 67 p.
- [48] B. Vauquois, S. Chappuy (1985)
Static grammars. A formalism for the description of linguistics models. International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, Colgate University, August 14-16, 1985, 25 p.

Other bibliographical references on various aspects of MT

- [49] Bachut Daniel, Verastegui Nelson (1984)
Software tools for the environment of a computer-aided translation system. Proc. of COLING-84, ACL, 330-334, Stanford.
- [50] Bar Hillel Yehoshua (1953)
Machine Translation. Computer and Automation 2(5), 1-6.
- [51] Bar Hillel Yehoshua (1971)
Some reflections on the present outlook for high-quality machine translation. In: Feasibility study on fully automatic high quality translation, RADC-TR-71-295, 73-76.
- [52] Boitet Christian (1976)
Un essai de réponse à quelques questions théoriques et pratiques liées à la traduction automatique. Définition d'un système prototype. Thèse d'Etat, Grenoble.
- [53] Boitet Christian (1985)
Traduction (assistée) par Ordinateur: ingénierie logicielle et linguicielle. Colloque RF&IA, AFCET, Grenoble.
- [54] Boitet Christian (1986a)
The French National MT-Project: technical organization and translation results of CALLIOPE-AERO. IBM Conf. on Translation Mechanization, Copenhagen.
- [55] Boitet Christian (1986b)
Current Machine Translation systems developed with GETA's methodology and software tools. ASLIB Conf. London.
- [56] Boitet Christian (1987a)
Research and development on MT and related techniques at Grenoble University (GETA). Revised from Boitet (1984), in "Machine Translation today: the state of the art", Proc. third Lugano Tutorial, 2-7 April 1984, M. King, ed., Edinburgh University Press, 1987, 133-153.
- [57] Boitet Christian (1987b)
Current projects at GETA on or about Machine Translation. Proc. II World Basque Congress, San Sebastian, 7-11 Sept. 1987, 28p.
- [58] Boitet Christian (1987c)
Current state and future outlook of the research at GETA. Proc. of the Machine Translation Summit, Hakone, 17-19 Sept. 1987, 26-37.
- [59] Boitet Christian (1987d)
Situation et évolution des recherches en TAO au GETA. Bulletin de la société franco-japonaise des techniques nouvelles, 33/2-3, 1987, 24-27.
- [60] Boitet Christian (1988a)
Software and lingware engineering in modern M(A)T systems. To appear in "Handbook for Machine Translation", Batori, ed., Niemeyer, 1988.
- [61] Boitet Christian (1988b)
Pros and cons of the pivot and transfer approaches in multilingual Machine Translation. Proc. of the int. conf. on "New directions in Machine Translation", BSO, Budapest, 18-19 August, 1988.
- [62] Boitet Christian, Gerber René (1984)
Expert systems and other new techniques in MT. Proc. COLING-84, ACL, 468-471, Stanford.
- [63] Boitet Christian, Nedobejkine Nikolai (1981)
Recent developments in Russian-French Machine Translation at Grenoble. Linguistics 19 (3/4), 199-271.
- [64] Boitet Christian, Nedobejkine Nikolai (1983)
Illustration sur le développement d'un atelier de traduction automatisée. Colloque "l'informatique au service de la linguistique", Univ. de Metz.
- [65] Boitet Christian, Nedobejkine Nikolai (1986)
Toward integrated dictionaries for M(a)T: motivations and linguistic organization. Proc. COLING-86, IKS, 423-428, Bonn.

- [66] Boitet Christian, Guillaume Pierre, Quézel-Ambrunaz Maurice (1978)
Manipulation d'arborescences et parallélisme: le système ROBRA. Proc. COLING-78, Bergen.
- [67] Boitet Christian, Guillaume Pierre, Quézel-Ambrunaz Maurice (1982)
ARIANE-78: an integrated environment for automated translation and human revision. Proc. COLING-82, North-Holland, Ling. series 47, 19-27, Prague.
- [68] Boitet Christian, Guillaume Pierre, Quézel-Ambrunaz Maurice (1985)
A case study in software evolution: from ARIANE-78 to ARIANE-85. Proc. of the Conf. on theoretical and methodological issues in Machine Translation of natural languages, 27-58, Colgate Univ., Hamilton, N.Y.
- [69] Boitet Christian, Zaharin Y. (1988)
Representation trees and string-tree correspondences. Presented for COLING-88, Budapest, 22-27 August, 1988.
- [70] Chandioux John, Guérard M.F. (1981)
METEO: un système à l'épreuve du temps. Meta 26(1), 17-22.
- [71] Chappuy Sylviane (1983)
Formalisation de la description des niveaux d'interprétation des langues naturelles. Thèse, Grenoble.
- [72] Chauché Jacques (1974)
Transducteurs et arborescences. Etude et réalisation de systèmes appliqués aux grammaires transformationnelles. Thèse d'Etat, Grenoble.
- [73] Chauché Jacques (1975)
Les langages ATEF et CETA. AJCL, microfiche 17, 21-39.
- [74] Clemente-Salazar Marco (1982)
Une structure intéressante en TA: les E-graphes. Proc. COLING-82, North-Holland, Ling. series 47, Prague.
- [75] Colmerauer Alain (1970)
Les systèmes-Q, un formalisme pour analyser et synthétiser des phrases sur ordinateur. TAUM, Univ. de Montréal.
- [76] Courtin Jacques (1973)
Un analyseur morphologique et syntaxique pour les langues naturelles. Proc. COLING-73, Pisa.
- [77] Ducrot Jean-Marie (1973)
Le système TITUS II. ANRT information and documentation 4, 3-40.
- [78] Ducrot Jean-Marie (1982)
TITUS IV. In: *Information research in Europe*. Taylor, P.J., Cronin, P., eds., Proc. of the EURIM 5 conf., Versailles, ASLIB, London.
- [79] Dymetman Marc (1986)
Two approaches to Common Sense Inferencing for Discourse Analysis. Proc. COLING-86, IKS, 511-514, Bonn.
- [80] Feng Zhi Wei (1981)
Mémoire pour une tentative de traduction multilingue du chinois en français, anglais, japonais, russe et allemand. Doc. GETA, Grenoble, 40p. + annexes.
- [81] Fillmore C.J. (1968)
The case for case. In: *Universals in linguistic theory*, Bach & Harms, eds, Holt, Rhinehart & Winston, N.Y., 1-88.
- [82] Gerber René (1984)
Etude des possibilités de coopération entre un système fondé sur des techniques de compréhension implicite (système logico-syntaxique) et un système fondé sur des techniques de compréhension explicite (système expert). Thèse, Grenoble.
- [83] Gerber René, Boitet Christian (1985)
On the design of expert systems grafted on MT systems. Proc. of the Conf. on theoretical and methodological issues in Machine Translation of natural languages, 116-134, Colgate Univ., Hamilton, N.Y.
- [84] Grishman R., Sager Naomi, Raze C., Bookchin B. (1973)
The Linguistic String Parser. Proc. of AFIPS-73, 427-433.

- [85] Gross Maurice (1986)
Lexicon-grammar: the representation of compound words. Proc. COLING-86, IKS, 1-6, Bonn.
- [86] Guilbaud Jean-Philippe (1984)
Principles and results of a German-French MT system. Lugano tutorial on Machine Translation.
- [87] Guilbaud Jean-Philippe (1986)
Variables et catégories grammaticales dans un modèle ARIANE. Proc. COLING-86, IKS, 405-407, Bonn.
- [88] Hagège Claude (1982)
La structure des langues. PUF, Paris.
- [89] Hajic^ova Eva, Kirschner Zdenek (1981)
Automatic translation from English into CzeChristian Prague Bulletin of Mathematical Linguistics 35, 5-23.
- [90] Harris Zellig (1962)
String analysis of sentence structure. Mouton & Co, The Hague.
- [91] Hauenschild Christa, Huckert Edgar, Maier R. (1979)
SALAT: machine translation via semantic representation. In: Semantics from different points of view, Bäuerle & al., eds., Springer, Berlin, 324-352.
- [92] Hutchins W.J. (1978)
Machine translation and machine-aided translation. Journal of Documentation 34(2), 119-159.
- [93] Hutchins W.J. (1979)
Linguistic models in Machine Translation. UEA papers in linguistics 9, 29-52.
- [94] Hutchins W.J. (1982)
The evolution of machine translation systems. In: Practical experience of machine translation, Proc. ASLIB-81 Conf., London, 5-6 Nov. 1981, Lawson, ed, North Holland, 21-37.
- [95] Hutchins W.J. (1986)
Machine Translation : Past, Present, Future. Ellis Horwood, John Wiley & sons, Chichester, England, 382 p.
- [96] Isabelle Pierre, Bourbeau Laurent (1984)
TAUM-AVIATION: its technical features and some experimental results. Computational Linguistics, 11:1, 18-27.
- [97] King Maghi, Perschke Serguei (1982)
EUROTRA and its objectives. Multilingua 1(1), 27-32.
- [98] King Maghi, Johnson Rod, Des Tombe Louis (1985)
EUROTRA: a multilingual system under development. Computational Linguistics, 11:2-3, 155-169.
- [99] McCord Michael (1985)
LMT: a Prolog-based Machine Translation system. Proc. of the Conf. on theoretical and methodological issues in Machine Translation of natural languages, 179-182, Colgate Univ., Hamilton, N.Y.
- [100] Lawson Veronica (1982), ed
Practical experience of Machine Translation. Proc. ASLIB-81 Conf., London, 5-6 Nov. 1981, North-Holland.
- [101] Lawson Veronica (1983)
Machine Translation. In: The translator's handbook, Picken, ed., ASLIB, London, 81-88.
- [102] Lawson Veronica (1985), ed
Tools for the trade, Translating and the computer 5. Proc. ASLIB-81 Conf., London, 10-11 Nov. 1983.
- [103] Lepage Yves (1986)
A language for transcriptions. Proc. COLING-86, IKS, 402-404, Bonn.
- [104] Luckhard H.D. (1982)
SUSY: capabilities and range of applications. Multilingua 1(4), 213-219.
- [105] Maas Heinz-Dieter (1981)
SUSY I und SUSY II: verschiedene Analysestrategien in der Maschinellen Übersetzung. Sprache und Datenverarbeitung 5(1/2), 9-15.

- [106] Melby Alan (1982)
Multilevel translation aids in a distributed system. Proc. COLING-82, North-Holland, Ling. series 47, 215-220, Prague.
- [107] Mel'chuk Igor (1981)
Dependency models: a recent trend in Soviet linguistics. Annual Review of Anthropology 10, 27-62.
- [108] Nakamura Junichi, Tsujii Junichi, Nagao Makato (1984)
Grammar writing system (GRADE) of Mu-Machine Translation Project and its characteristics. Proc. COLING-84, ACL, Stanford.
- [109] Nomura Hiroshiro, Naito Shozo, Katagiri Yasuhiro, Shimazu Akira (1986)
Translation by understanding: a Machine Translation system LUTE. Proc. COLING-86, IKS, 621-626, Bonn.
- [110] Petrick Stanley (1965)
A recognition procedure for transformational grammars. Ph.D. Thesis, Dep. Modern Languages, MIT, Cambridge, Massachusetts.
- [111] Sakamoto Yoshiyuki, Satoh Masayuki, Ishikawa Tetsuya (1984)
Lexicon features for Japanese syntactic analysis in MU-project-JE. Proc. COLING-84, ACL, Stanford.
- [112] Salkoff Morris (1973)
Une grammaire en chaîne du français. Dunod, Paris.
- [113] Salkoff Morris (1979)
Analyse syntaxique du français: grammaire en chaîne. Linguisticae investigaciones, Supplementa 2.
- [114] Sedogbo Célestin (1985)
Tree grammar: a Prolog implementation of the string grammar of French. Proc. COGNITIVA85-II, 893-896.
- [115] Slocum Jonathan (1984)
METAL: the LRC Machine Translation system. Lugano tutorial on Machine Translation.
- [116] Stewart Gilles (1975)
Manuel du langage REZO. TAUM, Univ. de Montréal.
- [117] Tomita Masaru, Carbonell Jaime G. (1986)
Another stride towards knowledge-based Machine Translation. Proc. COLING-86, IKS, 633-638, Bonn.
- [118] Tong Loong Cheong (1986)
English-Malay translation system: a laboratory prototype. Proc. COLING-86, IKS, 639-642, Bonn.
- [119] Tsujii Jun-Ichi (1987)
What is pivot? Proc. of the MTS, p. 121, Hakone.
- [120] Vauquois Bernard (1960-1985)
see complete list above.
- [121] Veillon Gérard (1970)
Modèles et algorithmes pour la traduction automatique. Thèse d'Etat, Grenoble.
- [122] Verastegui Nelson (1982)
Utilisation du parallélisme en traduction automatisée par ordinateur. Proc. COLING-82, North-Holland, Ling. series 47, 397-405, Prague.
- [123] Walker Donald E. (1985)
Knowledge resource tools for accessing large text files. Proc. of the Conf. on theoretical and methodological issues in Machine Translation of natural languages, 335-347, Colgate Univ., Hamilton, N.Y.
- [124] Wheeler Peter (1986)
The LOGOS translation system. Proc. of IAI-MT, 20-33, IAI-EUROTRA-D, Saarbrücken.
- [125] Wilks Yorick (1975)
Preference Semantics. Formal semantics of Natural Language, Keenan, ed., Cambridge, 329-348.
- [126] Witkam Toon (1983)
Distributed Language Processing: feasibility study of a multilingual facility for videotex information networks. Utrecht, BSO.

- [127] Witkam Toon (1987)
Interlingual MT – an industrial initiative. Proc. of the MTS, 135–140, Hakone.
- [128] Woods William (1970)
Transition network grammars for natural language analysis. CACM 13/10, 591-606.
- [129] Zaharin (1986a)
Strategies and heuristics in the analysis of a natural language in Machine Translation. Ph.D. thesis, Universiti Sains Malaysia, Penang (research conducted in cooperation with GETA, Grenoble).
- [130] Zaharin (1986b)
Strategies and heuristics in the analysis of a natural language in Machine Translation. Proc. COLING-86, IKS, 136–139, Bonn.
- [131] Zajac Rémi (1986a)
Etude des possibilités d'interaction homme-machine dans un processus de traduction automatique. Spécification d'un système d'aide à la rédaction en vue d'une traduction par machine. Définition d'un langage de spécification linguistique. Thèse, Grenoble.
- [132] Zajac Rémi (1986b)
SCSL: a linguistic specification language for MT. Proc. COLING-86, IKS, 393-398, Bonn.
- [133] Yang Ping (1981)
Un essai sur la génération du chinois. Doc. GETA, Grenoble, 30p. + annexes.
- [134] Yan Yong Fong (1987)
Vers une ingénierie de la production de linguiciels. Spécification et réalisation d'un prototype de poste de travail linguistique pour la spécification de correspondances structurales. Thèse, Univ. de Grenoble.
- [135] Yngve Victor (1957)
A framework for syntactic translation. Mechanical Translation 4(3), 59–65.

-o-o-o-o-o-o-o-o-o-