

PRESENT AND FUTURE PARADIGMS
IN THE AUTOMATIZED TRANSLATION
OF NATURAL LANGUAGES.

Ch. BOITET, P. CHATELIN, P. DAUN FRAGA
GETA, F-38041 GRENOBLE CEDEX 53X, FRANCE.

Abstract

Useful automatized translation must be considered in a problem-solving setting, composed of a linguistic environment and a computer environment. We examine the facets of the problem which we believe to be essential, and try to give some paradigms along each of them. Those facets are the linguistic strategy, the programming tools, the treatment of semantics, the computer environment and the types of implementation.

Introduction

Machine Translation has been a recurring theme in applied linguistics and computer science since the early fifties. Having not yet attained the enviable status of a science, it is best considered as an art in the same way as Knuth considers computer programming. Failure to recognize that MT must be treated in a problem-solving setting, that is, as a class of problems to be solved in various environments and according to various quality and cost criteria, has led and still leads to impassionate, antiscientific attitudes, ranging polemically between dreamy optimism and somber pessimism. Using the fairly large body of experience gained since the beginning of MT research, we try in this paper to extract the most essential facets of the problem and to propose some paradigms, alongside each of those facets, for usable computer systems which should appear in the near - or middle - term future.

As a matter of fact, the phrase "Machine Translation" is nowadays misleading and inadequate. We shall replace it by the more appropriate term "Automatized Translation" (of natural languages) and abbreviate it to AT.

Part I tries to outline the problem situations in which AT can be considered. The following parts examine the different facets in turn. Part II is concerned with the linguistic strategy, Part III with the programming tools, Part IV with semantics, Part V with the computer environment and Part VI with possible types of implementation.

I - Applicability, quality and cost : a problem situation.

1. The past

Automatized translation systems were first envisaged and developed for *information gathering* purposes.

The output was used by specialists to scan through a mass of documents, and, as RADC user report shows [49], the users were quite satisfied. This is no more the case with the growing need for the *diffusion of information*. Here, the final output must be a good translation. Second generation systems were designed with this goal in mind, and with the assumption that good enough translations cannot *now* be obtained automatically on a large scale, but for very restricted domains (see METEO). Hence, a realistic strategy is to *try to automate as much as possible of the translation process*. This is the approach taken by GETA, TAUM, LOGOS, PROVOH and many others. Here, the problem is to answer existing needs by letting man and machine work together.

Another approach comes from AI and is best exemplified in [9]. Here, the goal is more theoretical : how to simulate a human *producing* competent translations ? We will argue that the methods developed in this framework are not yet candidates for immediate applicability.

PARADIGM 1 : Future MT systems will be AT (automated) systems rather than completely automatic systems.

2. Applicability

Automated translation is clearly needed by large organizations as the EC or big industries having to translate tens or hundreds of millions of pages per year. Translations are very often urgent, and there is a human impossibility, as translations needs increase much faster than the number of available translators.

Translators are specialized to certain kinds of texts. In the same way, AT systems, which are costly to develop and maintain, should be tailored to some kind of texts : AT is applicable when there is a constant flow of very homogeneous and repetitive texts, hopefully already in machine-readable form. AT should allow to integrate automatic rough translation and human on- or off- line revision.

3. Quality

This is a crucial point, when it comes to acceptability by revisors and/or end-users. The quality of translation is a very subjective notion, relative to the need and knowledge of the reader. Traditional counts of grammatical errors, false senses, nonsenses give only indications.

We believe that quality should be estimated by the amount of revision work needed, to compare it with human (rough) translation, which is also very often of poor quality. As errors of AT systems are certain to be different from those of humans, revisors must have a certain training before such a comparison can be made.

Another measure could be to compare final translations, with the same amount of revision. We believe this not to be realistic, as cost must also be taken into account : translators will turn to revision, which is much faster than translation, so that they will gain time even if they work for revision.

4. Cost

The cost of AT should be divided into the costs of development, maintenance and use. It is of course related to the linguistic and computer environments. First, the graph of language-pairs should be considered, as development costs for an analyzer, say, may be charged to different pairs with the same source, of course *if analysis and synthesis are strictly monolingual*.

Easy maintenance calls for sophisticated computer systems having an interactive data-base aspect and concise metalanguages with good, incremental compilers.

Machine time tends to be a minor component of the cost of use. Important savings come from the *integration of the human revision in the AT system* (see TAUM, LOGOS, GETA), as no further typing is required.

5. Text typology

AT systems developed for simple texts will certainly be less expensive (and probably better) than those for complex texts. Let use give a tentative hierarchy. The easiest texts are perhaps already *preedited abstracts* regularly entered into data bases. Then come *abstracts*, which may present surprising difficulties mainly due to the tendency to write everything in one long and badly constructed sentence.

Technical documentation, maintenance manuals, etc. are more and more written in a systematic way, which permits to tailor an AT system to their form and their content. See however TAUM-AVIATION reports for a sobering view on their apparent facility ! *Minutes of meetings and working documents* may be still harder.

Newspaper articles, even on scientific subject matters, tend to accumulate difficult or incorrect constructions, and also to jump far away from the current subject matter.

Until AI methods ("third" or "fourth") generation are practicable with really large data, we don't believe AT systems should even try to handle literary, normative or diplomatic texts. Revision would just be a new translation.

PARADIGM 2 : AT systems are now applicable only in restricted environments and must be tailored to particular kinds of texts.

II - Linguistic strategy

1. Multilingual or pair-oriented systems ?

Almost all AT systems divide the process of translation in three main logical steps, analysis, transfer and synthesis. At one extreme, some systems (like METEO) are strongly oriented towards a particular pair of languages. This means that analysis of the source language is performed with the knowledge of the target language. Target lexical units may appear during analysis, and syntactic or semantic ambiguities in the source are handled contrastively.

The other extreme is the complete independence of analysis and synthesis. This is the approach taken in multilingually oriented systems (like ARIANE-78 [7, 36, 50], SUSY [51], SALAT [18, 20], TAUM/AVIATION [33]). This independence enhances modularity and economically justified, as analysis or synthesis are written once for each language. Analysis usually represents at least 2/3 of the programming effort and computing time.

PARADIGM 3 : We advocate for multilingually oriented systems, where the basic software itself guarantees independence of analysis and synthesis.

2. What kind of analysis ?

Should the analysis deliver a structural descriptor of the unit of translation, or a representation of its meaning, static or dynamic ? With the first approach, the transfer step includes necessarily a lexical transfer and a structural transfer. With the second one, the result of the analysis is a language-independent representation of the unit of translation (sentence, paragraph(s)). When the lexical units themselves are language-free, as in SAM [9], we call it "pure pivot" approach. When only the relations between units are language-free, we call it "hybrid pivot" approach (as in the first CETA [34, 35] system). In the first case, there is no transfer, in the second, transfer is purely lexical.

The pivot approach is theoretically very elegant. However, past experience with it (on a corpus of more than a million words, see Vauquois (1975)) shows that it is quite inadequate in real situations, where, very often, this representation cannot be obtained, or not for all parts of the translation unit. Also, human professional translators seem very often to produce quite acceptable results without actually abstracting the deep meaning and rephrasing it, but rather by using standard syntactic transformations (like active-passive, reordering of nominal groups, passive-impersonal, splitting up sentences, etc.) and ... multiple

choice bilingual dictionaries. If deep comprehension fails, it is hence necessary and possible to fall back on lower levels of information.

PARADIGM 4 : The result of analysis should be a structural descriptor of the unit of translation, where the lexical units are still source lexical units and where the linguistic information is "multi-level" : logical relations, syntactic functions, syntactic classes, semantic features (all universal for large families of languages), and trace information (proper to the source language).

As we argue in Part IV, we don't think the result of analysis should include a dynamic comprehension of "what is described to happen", at least in AT systems for the near future. Let us quote Carbonell & al (1978) : "What kind of knowledge is needed for the translation of texts? Consider the task of translating the following story about eating in a restaurant...". Unfortunately, the texts to be translated, as we said in Part I, are not stories, but rather abstracts, manuals, working documents ... of a very different nature.

3. Strategical aspects

There are some problems the analysis writer can not escape. Should problems such as ambiguities be solved as soon as they appear, or not be solved altogether, or is it better to devise strategies to decide as late as possible, or more complex heuristics ?

PARADIGM 5 : AT systems to be developed in the near future should allow complex linguistic heuristics. That is, we feel that preferences computed by the use of weights derived from some frequency counts are not enough, and that linguists should program what they see as being essentially heuristic in the linguistic processes. Hence further requirements on the programming tools, which should at least include such control structures as controlled non-determinism.

III - Programming tools : algorithmic models and metalanguages

1. History

The first MT researchers programmed directly in machine language. Until now, SYSTRAN systems are essentially made of programs and tables written in IBM 370 macroassembler. Then came systems based on a simple formal model, like context-free grammars and Q-systems. These systems rely on a general algorithm over which the rules have no control. Systems allowing such controls (PROLOG [52], ATEF [14, 15], ROBRA [50], ATNs [47] and derived models like REZO [32], PLATO, DEDUKT [18, 20]) were created in the seventies.

Now, the programming languages used to write the linguistic part of AT systems include

usual programming languages such as macro-assembler, FORTRAN, ALGOL, PL/1, LISP, as well as specialized languages (see above).

2. The need for powerful data and control structures

In our view, usual programming languages are inadequate as metalanguages to be used for writing the linguistic data and procedures in an AT system.

PARADIGM 6 : Adequate metalanguages should include built-in complex data-types such as decorated trees and graphs as well as control structures for non-deterministic, parallel and heuristic programming.

Note that parallelism may be of two different kinds : processors working independently on independent data structures and processors working on a common data structure (e.g. a normal context-sensitive grammar is not equivalent to the same grammar used in parallel, see $S \rightarrow AB, A \rightarrow a/-B, B \rightarrow b/A-$). Many recent specialized programming languages include a form of non-determinism, but very few have parallelism (ROBRA) or control functions for heuristics (PROLOG, ATEF, REZO).

Of course, these metalanguages should include more classical control structures such as iteration, recursion or selection. Note that dictionaries are simply big "select" constructs, possibly non-deterministic (one-many).

3. Complexity, decidability, adequacy

If one takes all necessary data-types with all possible operators and all control structures, the model obtained is very likely to have the (maximal) computing power of a Turing machine. Hence, no general bound or estimate for the dynamic complexity of programs written in that formalism may be given. On the other hand, as all we want to program in AT systems is certainly subrecursive, another approach is to define several subrecursive algorithmic models with associated known (or studyable) complexity classes. This was the original approach at GETA, with the ATEF, ROBRA, TRANSF and SYGMOR algorithmic models, designed to be decidable and of linear complexity.

As a matter of fact, decidability is a very practical requirement. However, general constraints imposed to guarantee decidability may make certain things unnecessarily clumsy to write. Perhaps a better idea (implemented in ATNs, ATEF and ROBRA) is to build algorithmic models as extensions of decidable models, in such a manner that sources of undecidability are easy to locate, so that particular decidability proofs may be looked for. For example, the fundamental operator of ROBRA is the parallel application of the rules of a transformational grammar to an object tree.

Normal iteration of this operator must terminate, due to some marking mechanism. However, a grammar in "free" iteration mode may never terminate.

Last, but not least, these algorithmic models must be adequate, in the sense of ease and concision of writing. We sum up with

PARADIGM 7 : The complex operators associated with the data types should be adequate, their complexity (time and space) should be reasonably bounded ($O(n)$ to $O(n^3)$?) and there should be decidable underlying algorithmic models, so that sources of undecidability could easily be traced.

IV - Semantics

1. Two different notions

Semantics are understood differently in linguistics, logic and computer science. In the latter, attention is focused on the ways of expressing data and processes. A system is said to be "syntactic" if it operates within the framework of formal language theory, that is by combinatorial processes on classes or "features".

In a "static" semantic system, there is a fixed model of some universe, possibly represented as a thesaurus, or as a set of formulae in some logic, on which a formal language is interpreted.

A system incorporates "dynamic semantics", or "pragmatics", if the interpretation of the data it processes may alter the model of the universe, or create a parallel model of some "situation" in this universe.

2. Classical approaches

Existing AT systems of reasonable size, that is incorporating several thousands of lexical units and quite exhaustive grammars, rely essentially on semantics by features. They may be quite refined and specialized to a domain (e.g. METEO), and, in that case, this method may give surprisingly good results.

Although the basic softwares allows to relate lexical units by using (monolingual or bilingual) dictionaries, this possibility is hardly used in the current applications at TAUM (see TAUM/AVIATION) or at GETA (see [50]). For instance, classical relations such as antonymy, generalization, particularization are not coded in the dictionaries.

3. AI proposals

AI proposals fall into two classes. The first refers essentially to static semantics, and may be illustrated by Wilks' "preference semantics" [37-44] or Simmons "semantic networks" [30].

As applied to AT, these methods have only been incorporated in very small size test programs incorporating at most some hundreds lexical units. However, we feel that their simplicity and relative economy in coding effort make them usable in near-term AT systems, under the essential condition that, as in Wilks' model, it is not necessary to code completely every lexical unit, and that the associated computing effort is controlled by the linguist and undertaken only when necessary, that is when a problem (like ambiguity or anaphoric reference) has not been solved by simpler means.

The second class of AI proposals relates to dynamic semantics and originates in the "frames" proposed by Minsky [12], and now proposed by other teams as "scripts", "plans" or "goals" [9, 27-29]. They are certainly very attractive, but have been demonstrated on very particular and very small size situations.

As we said above, texts to be translated with AT systems are more likely to be technical documents, abstracts, instructions for use, maintenance manuals, etc., than stories about restaurants or earthquakes. Each text doesn't rely on one clear-cut "script", or "type of situation", known a priori. Rather, such texts very often don't describe situations (see a computer manual), or, at the other extreme, their content might be understood as ... the description of hundreds of scripts (see aviation maintenance manuals).

Hence, our objection to the use of such methods is twofold. First, the coding effort, in principle, would be enormous. Charniak's frame for painting [12], although admittedly incomplete, is 19 pages of listing long (in a high-level language !), and we suppose he spent rather a long time on it. Just think of what it would cost to code 5000 basic frames, which we believe would be reasonable for, say, the domain of computers. Second, if the texts describe types of situations, then it is necessary to understand these texts in order to code the necessary scripts, which will be used ... to understand the text again !

This circularity has two consequences. First, only very general scripts might be humanly coded by using general previous knowledge about the domain. Second, if we want to use such methods extensively and at levels of detail at which they begin to give better results than simple approaches, then AI researchers should provide methods for the automatic extraction of scripts or frames from large bodies of texts, in an efficient (and perhaps interactive) way. That is, the use of such methods on wide domains and large amounts of texts entails automatic learning. Another problem is to automatically find which script is relevant to the current portion or text.

4. Concluding remarks

As in other problem-solving situations, simple methods should continue to be used in conjunction with more sophisticated ones. Unfortunately, proponents of "very high" semantics seem too often to concentrate on interesting high level phenomena as anaphoric reference, discourse structure, causality and reasoning and to forget at the same time persisting and very frequent lower-level difficulties such as ambiguity of prepositional group dependency. Take Riesbeck's [25 p. 11] example "John hurt Mary because Mary informed Bill that John advised Rita to prevent Bill from buying the book by giving the look to John". It seems obvious to the author that "by giving" relates to "prevent". However, it could also relate to "hurt", "informed" and "advised" ("buying" being excluded because of the coincidence of reference due to "the"). Take also Vauquois' example "the minister spoke of the candidate to the presidency", and many other occurring with conjunction and enumeration.

PARADIGM 8 : Reasonably large scale AT systems can rely only on semantics by features and static semantics in the near future. Script like methods must be completed by automatic script generation and retrieval procedures before they can be used extensively. Semantic methods must complete and not discord previous ones.

V - Computer environment for the users

1. Essential functions and types of users

There are different kinds of users of AT systems, intervening in different ways for different purposes, related to the functions of creation, maintenance and use. Specialized linguists create the linguistic systems or subsystems, lexicographers and terminologists create and update the dictionaries, revisors and translators use the system as a tool to produce translations, and the end user wants to know nothing about it, but is nevertheless the final judge.

PARADIGM 9 : A modern AT system must allow large degrees of interactivity at all functional levels, be transparent to the user, contain a (possibly specialized) data-base management system (for handling grammars, dictionaries, texts and their different versions as well as intermediate results and statistical information) and be integrated (from preediting and/or checking to revision and (photo) composition).

2. Types of use

At creation time, interactivity is certainly essential, even during parts which will be fully automatic in a production environment.

Perhaps a mode should be provided in which the system could ask simple questions (choice of equivalents, for instance) to a translator sitting at a screen while doing the automatic rough translation part. Even that may be too costly in a production environment.

For maintenance and updating of the linguistic data, we believe it is essential that an AT system provides ways of feed-back and communication between the different kinds of users, and between users and the system.

3. Human aspects

The human and social aspects should not be neglected. To force a rigid system on revisors and translators is a guarantee of failure. It must be realized that AT can only be introduced step by step into some preexisting organizational structure. The translators and revisors of the EC did not only reject Systran because of its poor quality but also because they felt themselves becoming "slaves of the machine", and condemned to a repetitive and frustrating kind of work.

PARADIGM 10 : AT systems must be such that the users keep control over them, and not vice versa.

VI - Types of implementation

This section is voluntarily short, as this question is not particular to AT systems. Hardware requirements for large MT systems are already met by minicomputers like IBM's 4300 series. Software requirements such as time-sharing and virtual memory are also available. Typically, GETA's current prototype Russian-French translation system executes with a central memory less than 1,5 Mbyte without any disk-access during the translation process, and uses 12 Mbytes on disk for linguistic files and the text data-bases. If the dictionaries would increase in size to, say, 40000 lexical units (more general than words, or roots), than 3 Mbytes of virtual memory and 20 to 25 Mbytes on disk would be needed. Even microcomputers might support such systems in the (longer term) future.

For the time being, such systems may be centralized, and operate on big computers, or be distributed on minicomputers, possibly linked through a network. The machine may be dedicated or not, and so forth. In addition to the hardware side, the software side is also important. Portability and efficiency are often conflicting goals. The implementation language(s) (not the metalanguages) may be low-level (assembler, LP language) or high-level (FORTRAN, PASCAL, PL/1, ALGOL68, LISP, ADA,...). Another possibility is to devise a special abstract machine for the metalanguage.

We believe that only the last two solutions should be considered, with *real* portability and efficiency as the main criterion for choosing a high-level language.

As a likely development, we foresee the use of AT systems first on big computers in large organizations, with or without teleprocessing, and then, in bureautics, on local minicomputers. However, some recent experience indicates that local development of user-tailored applications may well be done before bureaucratic inertia in large organizations allows decisions to be taken.

Acknowledgments

We would like to thank Pr. Vauquois as a principal inspirator of this work, although the errors are of course ours, and although his ideas on some points are certainly more advanced than those exposed here in the framework of possible near-immediate large scale applications.

B I B L I O G R A P H Y

- [1] Y. Bar-Hillel (1960), "The present status of Automatic Translation of languages" in : Advances in Computers, VI, 91-163.
- [2] Y. Bar-Hillel (1967), "Die Zukunft der maschinellen Uebersetzung, oder : warum Maschinen das Uebersetzen nicht erlernen", Sprache im technischen Zeitalter, 23.
- [3] Y. Bar-Hillel (1971), "Some reflections on the present outlook for High Quality Machine Translation", in : Feasibility study on fully automatic high quality translation, (RADC) Univ. of Texas.
- [4] Ch. Boitet (1976), "Méthodes sémantiques en TA", TA-informations n° 1, 3-42.
- [5] Ch. Boitet (1976), "Un essai de réponse à quelques questions théoriques et pratiques liées à la traduction automatique. Définition d'un système prototype". Thèse d'Etat, Univ. de Grenoble.
- [6] Ch. Boitet (1976), "Problèmes actuels en traduction automatique. Un essai de réponse", COLING-76, Ottawa, Preprint n° 33.
- [7] Ch. Boitet (1977), red., "Où en est le GETA début 1977 ?", Comm. groupée, 3ème congrès européen "Franchir la barrière linguistique", Luxembourg, p. 599-636. Also available in English and German.
- [8] Ch. Boitet (1977), "MT and the problem of understanding natural language", colloque franco-soviétique sur la TA, Moscou, déc. 1977.
- [9] J. Carbonnel, R.E. Cullingford, A.V. Gershman (1978), "Knowledge based Machine Translation, RR # 146, AI Project, Yale Univ., and Proc. COLING-78, Bergen.
- [10] E. Charniak (1975), "A partial taxonomy of knowledge about actions", ISSCO, WP # 13, Castagnola.
- [11] E. Charniak (1975), "Organization and inference in a frame-like system of common sense knowledge", ISSCO, WP # 14, Castagnola.
- [12] E. Charniak (1978), "A framed painting-representation of a common-sense knowledge fragment", Cognitive Science, V2, N1.
- [13] J. Chauché (1974), "Transducteurs et arborescences. Etude et réalisation de systèmes appliqués aux grammaires transformationnelles". Thèse d'Etat, Univ. de Grenoble.
- [14] J. Chauché (1975), "The ATEF and CETA systems", TA-informations n° 2 & AJCL, Microfiche 17, 21-39.
- [15] J. Chauché, P. Guillaume, M. Quézel-Ambrunaz (1973), "Le système ATEF", Doc. GETA G-2600-A, Univ. de Grenoble.
- [16] A. Colmerauer (1971), "Les systèmes-Q, ou un formalisme pour analyser et synthétiser des phrases sur ordinateur", in : TAUM-71, Univ. de Montréal.
- [17] R.W. Floyd (1967), "Non-deterministic algorithms", JACM, VI4, N4, 636-644.
- [18] C. Hauenschild, E. Huckert, R. Maier (1978), "SALAT : Machine Translation via semantic representation", in : Bäuerle & al.
- [19] Th. Hoffman (1978), "Semantics in aid of automatic translation", COLING-78, Bergen.
- [20] E. Huckert (1979), "Automatische Synthese des Französischen aus einer logischen Basis", AQ-Verlag, Dudweiler, Saarland.
- [21] R. Kittredge, L. Bourbeau, P. Isabelle (1978), "Design and implementation of an English-French Transfer Grammar", COLING-78, Bergen.
- [22] N. Nedobejkine (1976), "Niveaux d'interprétation dans une traduction multilingue : application à l'analyse du russe", COLING-76, Ottawa.

- [23] R. Quillian (1968), "Semantic memory", in : Semantic information processing, MIT Press, 216-270.
- [24] R. Quillian (1969), "The teachable language comprehender : a simulation program and a theory of language", CACM, V12, N8, 459-476.
- [25] C.K. Riesbeck (1974), "Computational understanding : analysis of sentences and context", ISSCO, WP #4, Castagnola, 241 p.
- [26] E. Sandewall (1971), "Representing natural language information in predicate calculus", in : Machine Intelligence 6, Meltzer & Mitchie, ed., American Elsevier.
- [27] R.C. Shank (1973), "Identification of conceptualizations underlying natural language", in [48], 187-247.
- [28] R.C. Shank (1974), "Understanding paragraphs", ISSCO, WP #6, Castagnola.
- [29] R.C. Shank, C.J. Rieger III (1974), "Inference and the computer understanding of natural languages", Artificial intelligence 5, 373-412.
- [30] R.F. Simmons (1973), "Semantic networks : their computation and use for understanding english sentences", in [48], 63-113.
- [31] R.F. Simmons, J. Slocum (1972), "Generation of english discourse from semantic networks", CACM, V15, N10, 891-905.
- [32] G. Stewart (1975), "Le langage de programmation REZO", TAUM, Univ. de Montréal.
- [33] TAUM (1979), "Présentation de la chaîne de traduction informatisée TAUM/AVIATION", Univ. de Montréal, 27 mars 1979.
- [34] B. Vauquois (1968), "Structures profondes et traduction automatique. Le système du CETA", Revue Roumaine de linguistique, 13, 105-130.
- [35] B. Vauquois (1975), "La traduction automatique à Grenoble", Doc. de Linguistique Quantitative n° 24, Dunod, 184 p.
- [36] B. Vauquois (1976), "Automatic translation - a survey of different approaches", COLING-76, Ottawa, and SMIL 1, 127-135.
- [37] Y. Wilks (1968), "On-line semantic analysis of english texts", Mechanical Translation, V11, N3 & 4, 59-72.
- [38] Y. Wilks (1972), "Grammar, meaning and the machine analysis of language", Routledge London, 198 p.
- [39] Y. Wilks (1973), "An artificial intelligence approach to machine translation", in [48], 114-151.
- [40] Y. Wilks (1973), "Preference semantics", Stanford AI lab., AIM-206, CS-73-377.
- [41] Y. Wilks (1975), "An intelligent analyzer and understander of english", CACM, V18, N5, 264-274.
- [42] Y. Wilks (1975), "Seven theses on artificial intelligence and natural language", ISSCO, WP #17, Castagnola.
- [43] Y. Wilks (1976), "A preferential, pattern-seeking, semantics for natural language inference", Artificial Intelligence 6.
- [44] Y. Wilks & M. King (1976), "Semantics, Preference and Inference. A full description of a system and a program", ISSCO, WP #18, Geneva.
- [45] T. Winograd (1971), "Procedures as a representation for data in a computer program for understanding natural language", AI-TR-17, MIT.
- [46] T. Winograd (1973), "Procedural model of language understanding", in [48], 152-186.
- [47] W.A. Woods (1975), "Syntax, semantics and speech", BBN report n° 3067, AI report n°27.
- [48] Shank & Colby, ed. (1973), "Computer models of thought and language", Freeman & Co, San Francisco.
- [49] Z.L. Pankowicz, technical evaluator (1973), "User's evaluation of machine translation Georgetown MT system (1963-1973)", RADC-TR-73-239, Univ. of Texas at Austin.
- [50] C. Boitet, N. Nédobejkine (1980), "Russian French at GETA : outline of the method and detailed example", to appear in the Proc. of COLING-80, Tokyo.
- [51] H.D. Maas (1978), "Das Saarbrücker Uebersetzungssystem SUSY", Sprache und Datenverarbeitung, 43-62.
- [52] A. Colmerauer, H. Kanoui, M. Van Caneghem (1979), "Etude et réalisation d'un système PROLOG", Groupe d'Intelligence Artificielle, Univ. d'Aix-Marseille II.

Note : EC European Community
AI Artificial Intelligence