MT in the 1970s and 1980s

Doug Arnold, Louis des Tombe and Steven Krauwer

April 2007

This is a transcript of an interview between the Interviewer and Doug Arnold, Louis des Tombe and Steven Krauwer, which took place in Utrecht, Geneva, and Wivenhoe in 2006, intended to get their recollections and reflections of Machine Translation in the 1980s.

Scene Settings

Welcome all of three of you, thank you for sparing the time for this interview. To start, how did you get into MT?

(SK) Well I guess it is a slightly different answer for each of us, but essentially for all of us, it was through the Eurotra project. This was the MT project that was created by the European Commission in the late 1970s. The aim was to produce a high quality, fully automatic machine translation system for use by the EC. The EC had a serious translation problem because of the number of languages involved (then six or seven, but predicted to soon grow to nine when countries like Spain, Portugal and Greece joined).

Didn't the EC already have an MT system?

(SK) Yes, they had SYSTRAN in use, which they were using and further developing for language pairs, but I think they thought it was too difficult to extend to new language pairs. Also, I think the EC, or at least, some people in the EC, wanted to develop expertise in Natural Language Processing within Europe.

Was the ALPAC report a factor here?

(DA) Not much. However devastating it had been for MT funding in the US, it was based on a number of

presuppositions that did not at all apply to the European context. The American situation involved mainly translation of scientific documents from Russian into English. Unlike the situation in Europe the American problem was not a 'multilinguality problem'. Furthermore ALPAC was based on a cost-benefit analysis of MT, and both the costs and the potential benefits had changed dramatically in the meantime.

An important scientific issue in ALPAC had been the 'Bar-Hillel problem' -- the problem of the 'AI completeness' of MT. That problem was still there, just as much in Europe as anywhere else, but we thought it might not be fatal. We thought we could get a long way on a purely linguistic basis, or at least, it was an empirical question how serious the problem would be in a real life situation. We were dealing with fairly closely related languages, a relatively limited domain (European Commission documents), and a system to be used in the context of the translation services of the European Commission, where post-editing was common practice.

Were you involved in the project from the beginning?

(LdT) No. Steven joined first, but we all joined at about the same time, around end 1980. This was when the founders of Eurotra started to look for people who could represent the languages involved. The reason we joined was that it looked like an interesting intellectual and scientific adventure: it seemed to offer a chance to really use linguistics for something, a challenge for contemporary theories of linguistics and computation. I used to say that MT should make linguistics respectable, like the atomic bomb had done for physics.

(DA) I think the international dimension was also very appealing.

(SK) The job market was also an important motivation: the job market for linguists with a formal or computational orientation was shrinking. At least in the Netherlands, it was becoming less easy for such students to get jobs in teaching, and MT seemed to offer a novel career path for them.

Do you all three have the same background?

(LdT) Steven's background is in mathematics and mine in psychology, we were working in the faculty of arts at Utrecht university. We were working together on the application of finite state transducers in linguistics, and we were teaching computational linguistics.

(DA) I started out in English Literature, but when I first encountered the project I was a theoretical linguist, without much expertise in computational linguistics.

Life within the Project

What was the project like when you joined it

(SK) My first real encounter with the project was a workshop in Bangor in Wales, which I was invited to attend. There were about 80 participants from all countries of the (then) European Community. It was a mix of linguists, computer scientists, translators, and some AI people. There were actually very few people with a strong background in MT. Some interesting people I remember from Bangor are Bernard Vauquois, Antonio Zampolli, Yorick Wilks, Dieter Maas, Jan Landsbergen, Maghi King, Rod Johnson and Serge Perschke.

My first impression was that it was all about linguistics. Most of the discussions were about representation of linguistic phenomena, though there was some discussion about system design, but that was mainly a smaller group. And most of the documentation the project had produced was about describing linguistic phenomena in the representational framework that was being developed. Most of the work was aimed at improving this framework and at dealing with representational problems in individual languages. There was a document called 'ET-199', mainly written by Maghi King, which set out the common linguistic ideas. It was based on dependency grammar, and inspired by the work from the GETA group in Grenoble.

(LdT) On coming back from Bangor, Steven and I discussed joining the Flemish-Dutch group that was being created to deal with the Dutch language. The first thing I did was to read this ET-199 document. It shocked me, because I had not seen linguistics like that before, especially the absence of motivations for proposals, and the lack of pointers to the existing literature. I thought this should be improved on. We got a small ‘study contract’ from the CEC to investigate the representation of Coordination and Comparison. At the same time the Danish group had a study contract on Time and Tense, and the Brits had one on the use of trace. Which I think you worked on, Doug.

(DA) Yes, I was at UMIST at the time, and made a small contribution to the study contract on the use of traces. As a result, I went to a training workshop in Leuven where I first met Louis, and then to a workshop in Urbino which involved everyone in Eurotra at the time. After this I moved to Essex, which was also involved in the project, because of Yorick Wilks.

It is interesting that the main training that was given in Leuven to new project members related to the use of

dependency grammar to represent linguistic structures -- it was an introduction of a representational system. The choice of dependency grammar was a reflection of the influence of GETA, which was the most advanced MT group in Europe at the time, even though dependency grammar was not a particularly fashionable framework in linguistics at that time.

So the training didn't involve an introduction to machine translation or computation?

(DA) Correct, and this reflects the way the project was seen as being mainly linguistic in nature. The Eurotra system was designed to be a transfer system. Text was analyzed to and generated from a representation called 'Interface Structure' which was input to and output from transfer. The task of the linguists was to write grammars for their own languages in such a way that would be possible to manage with 'simple transfer' -- that is, the only thing the transfer component would have to do was to replace the lexical items. This meant that an important task during the preparation of the project was to develop guidelines for the Interface Structures. The initial focus of linguistic research was on phenomena that were thought to be problematic, and groups were given study contracts on coordination, time, modality, etc.

So at this stage of preparation there was some linguistic research done; I suppose there was also some preparation work on software?

(SK) Definitely. There were many ideas and opinions - one idea was that each group would develop its own software - but the official doctrine of the project quickly became the idea that there would one single software basis for all languages, and that there should be a clear separation of software and linguistics -- algorithms and data, you might say. The software was supposed to be a controlled production system, very much inspired by TAUM METEO. The system was to be modular, where modules would have strict input-output characteristics, defined declaratively. So, if an input did not meet the required specification, the 'expectation', it would be simply ignored by the module; and the only object that could be returned were ones that met the declaratively defined output criteria, its 'goals'.

The software design put a strong emphasis on declarativeness. It was foreseen that there would be a control component, defined in terms of regular expressions. The idea was that the linguist would write production rules, which would then be combined into modules with defined expectations and goals, and that the interactions between rules or modules would be described by means of ‘control expressions’. This design was proposed by the team called the 'software group', in which Rod Johnson from UMIST had a strong influence. The approach was

not uncontroversial in the project -- for example some people thought that linguists should code their grammars in a conventional programming language like Pascal.

(DA) So there was a coherent idea about the structure of the system and the organizational structure: organizationally there was one central software team, because the software was to the same for everyone; and a collection of more or less autonomous language groups whose job was to produce the grammars and transfer rules. To ensure coherence among the language groups a 'coordination group' was established. This met every month or so, in Geneva at first, later in Luxembourg.

(LdT) One thing that was missing at this time was a central 'linguistic' team to parallel the central software team to coordinate the linguistic side of things. In response to this lack, the project leader Serge Perschke set up a small central linguistics team including Doug, Lieven Jaspaert (from Leuven), and myself. The task of this group was to write 'linguistic specifications' -- to define the representational framework, the different levels of representation that would be used by the language groups.

What did you think were the biggest problems at the time?

(LdT) My point of view, as a new member of the central linguistics 'specification group', was that existing linguistic theory did not provide answers to many of the important questions of representation. For example there were no clear and widely accepted answers to questions like how to represent coordination, anaphora, tense, thematic relations, and modality. So defining linguistic specifications could not just be a question of writing down the accepted linguistic wisdom and thereby automatically ensure that the language groups would be able to describe their languages in a way appropriate for simple (i.e. just lexical) transfer.

(SK) The central software team and the central linguistic team had slightly different ideas about how to resolve this problem. The linguistic team had the idea of imposing a strict separation, built into the system design, of research and development. The idea was to do transfer on the basis of syntactic representations (the 'feasible' representation), as consensus amongst linguists about this seemed to be essentially possible. As a consequence transfer could then no longer be guaranteed to be 'simple' so that consensus about linguistic representations would not be absolutely necessary. At the same time representational issues resulting from empirical data about transfer complexity would be the focus of research on the 'adequate' representation.

(LdT) The idea of the software group was slightly different -- they suggested adopting an existing linguistic and computational theory, specifically LFG. This would allow the project to focus more on development aspects and would have allowed us to exploit a lot of existing and on-going linguistic research in LFG.

(DA) These ideas were compatible -- LFG could be seen as providing a starting point for (`feasible') linguistic representations as well as a software engine for computation. But neither turned out to be acceptable to the project management. One reason for this was an understandable reluctance to depart so radically from what was seen as the project's foundations -- in particular, abandoning the principle of simple transfer, at least in the short term. Another was an equally understandable suspicion about the stability and applicability of mainstream linguistic theories like LFG.

(SK) This tension between research and development had been present in the project from the beginning. Groups had different expectations about the project: some of the groups had seen it as simply or at least mainly a development project where they would just apply their existing knowledge and techniques in a wider domain, but others felt that a significant amount of linguistic research would be required to meet the project goal of building a high quality MT system of advanced design -- as the official EC council decision leading to the project had put it.

(LdT) The rejection of these ideas from the central software and linguistic teams lead to some conflict and disagreement. What emerged from this was a proposal for an alternative, new framework for Eurotra, called initially the CAT framework, where computational and linguistic aspects were well integrated. This was very different from what we had before. The grammar system was very declarative, but without explicit control. Interface structures were defined independently in terms of 'atoms', more or less lexical items, and ‘constructors’, which built structures. That was the 'A' and the 'C' part of the name, and it fulfilled a role similar to 'expectations' and 'goals' in the previous system design. The 'T' part was 'translators', which mapped between the structures.

The motivation for this was not based on theory of translation, as there was no suitable candidate. It was based on a concern about making life not too difficult for the rule writers -- key ideas were maintainability, simplicity, declarativeness.

Translation was seen as a sequence of simple steps between representation languages, each of which had its own independent definition in terms of constructors and atoms. We felt it was crucial that the relation between the levels of representation should be very simple and systematic, so we

tried to apply the idea of 'compositionality', influenced by the Rosetta project, which used compositionality in a more principled way, but a way that was not feasible for Eurotra.

(SK) With this new design for the project, the essential conflict between research and development did not really go away of course, and there still many linguistic phenomena for which there existed no standard solution.

So apart from the R and D conflict, what other problems were there?

(DA) Software. The software was very slow. A major factor here was the stratificational design, i.e. the number of linguistic levels that had to be translated to each other. This caused a huge amount of non-determinism. The idea was that a surface constituent structure was produced first; then mapped to a representation of surface grammatical relations like subject and object, which was then mapped to an 'Interface Structure', based on something very similar to predicate-argument structure. In practice, this led to a huge number of constituent structures being produced, most of which were simply discarded in the next step. This slowed things down very badly.

(SK) Another factor was that all rule interpreters were written in Prolog. Of course Prolog is very elegant, and very good as a specification language, and for the kind of rapid prototyping which we thought was important; but it is too slow for operations of this complexity on the kind of hardware we had at the time – it was not suitable for a development system. Rapid prototyping for the programmers turned out resulted in slow prototyping for the rule writers. We got significant improvement when we were able to move to compiled Prolog, and later on when more powerful and faster machines became available.

(DA) The slowness of the software was a problem in two ways. One thing is that linguists have to test the rules they have written, and this is very difficult if a test run takes hours or even days. It also leads to the grammatical analyses being infected by what are essentially hacks intended to speed things up.

(LdT) An entirely different problem was that there was a huge diversity in the scientific backgrounds of the people working in the project. This made it difficult for people to understand each other, let alone for people to agree on matters. This was such a serious and recurring problem that we even coined an abbreviation for it: the DTSB problem -- Differences in Training and Scientific Background. This meant that even relatively simple proposals often produced very long discussions, based on different understandings.

(DA) You should also remember that there were enormous practical problems at the start: when the project started in the 1980s many of the groups did not have email yet, and there were incompatibilities between machines and software. I remember that communication within the project improved a lot when we got access to Eurokom -- an early electronic conferencing system.

How did the project work?

(SK) The bulk of the development work in the project (i.e. the writing of grammars and transfer modules) was carried out by a consortium of 'language groups'; one per language, each normally based in one or more universities in the EU member states. Initially there were seven language groups, spread over some fifteen centres, and even more when Spain, Portugal and Greece joined at the end of the 1980s.

(DA) The idea was that the software and the linguistic specifications were provided centrally, and language groups would use them to write grammars for their own languages and transfer rules into their own language. The idea that they would be responsible for transfer rules *into* their language was based on the traditional idea that translators normally translate into their own language. It may not have been the best idea, and it might have been better for groups to have been in charge of transfer in both directions, for example.

From the language group point of view, the crucial bureaucratic step in its creation was the signature of a contract between their national government and the EC, because only when this was in place could real work began. Naturally some governments moved more quickly than others, which caused serious organizational and synchronization problems. But as the contracts were signed the language groups began to employ people to work on the project.

(LdT) The main problem with this organisation was to get all the groups to agree on fundamental technical issues, in the interest of simple transfer. The main forum for the necessary discussions was the 'Liaison Group', which consisted of representatives of the language groups, and which took decisions about everything ranging from practical organizational issues, e.g. “where will the next annual workshop take place?” to detailed decisions on technical matters, in particular on how to coordinate representations, e.g. how to represent coordination, and what priorities to give to different linguistic problems. I remember that we took a vote on the question of whether 'scope' was an important problem for the project. An example was 'your free second cup of coffee' which should not translate to French as 'votre deuxième tasse de café gratuite’. The result of the vote was negative, so the problem was decided to be minor, which

amazed me very much at the time. Maybe I was somewhat too idealistic concerning the problems the project should try to address.

Coordinated by the liaison group, the groups worked essentially independently. No detailed coordination of rule writing was possible in such a large number of centers, the job had to be done implicitly by means of the software and the linguistic specifications.

(DA) At various stages of the project's lifetime there existed central special-purpose groups. In the early part of the project a lot of preparatory work was done by a group at ISSCO in Geneva, led by Maghi King, which acted as general support team and project secretariat, and as the first contributors to system design and linguistic specifications. In fact, for a long time ISSCO was the main hub of the whole project. Other groups of importance for some time were the software team and the linguistic specification team (the two joined for a while to form 'the central team'). Later on there was a task force for issues of lexicography, a planning committee, and special contracts for groups in Liege, Luxembourg, and Dublin, who didn't have their own language, to work on terminology and lexicography.

You just mentioned the annual workshop -- that was maybe the main instrument we used to overcome the DTSB problem. Every summer essentially everyone who was working on the project, sometimes nearly 200 people, would get together in one place to discuss new ideas in small working groups. Socially this was very important too, and it gave people a sense of being part of a large project.

What was it like to work in a language group?

(DA) A typical language group was somewhere between ten and twenty people, based in a University, mostly in a Linguistics department. The main occupation of a language group member was to write grammars for analysis, generation and transfer, according to the linguistic specifications developed centrally. This could be pretty tedious work, and often frustrating given the slowness of the software.

I think a major problem for language group members was that they were employed by Universities as researchers, but actually doing development work, for which they got no academic credit. We took this problem very seriously at Essex, and constantly made efforts to find ways in which people could combine their development work with work that could lead to academic results. I don't know how successful we were, but it was something we tried to do.

What were your own roles in the project?

(SK) Each of the three of us participated in central teams at times and we were leaders of our language groups.

Outside the Project

References at the end of the interview.

What were the main intellectual and cultural influences on your work?

(DA) For me, Generative grammar and mathematical logic/semantics. The same for Louis and Steven, with the addition of automata theory. We had almost no influence from translation studies, or AI.

What did you think about other projects and researchers at the time?

(LdT) The 'direct' approach to MT, as exemplified by SYSTRAN, with no linguistic theory behind it, we thought was doomed to failure, because these systems would be impossible to maintain or extend.

(SK) Running at the same time as Eurotra, work was done on the METAL system, and our impression, based on what was said by people in that project, was that the approach was very similar to Eurotra.

(LdT) Rosetta was very elegant in design and realistic, we admired it very much. It was visionary, but unfortunately it was not a feasible design in a distributed project such as Eurotra, because it seemed impossible to produce isomorphic grammars in groups working independently.

(DA) There was a lot of activity in Japan, and we had several meetings with people from Japanese MT teams, but we found it difficult to see what the fundamental design ideas were, beyond the conventional view of a transfer based system. This was before Nagao proposed the idea of example based translation.

(SK) DLT was very visionary, but the initial vision was wrong: using Esperanto as the interlingua seemed to us not to solve any problems, but just to add to them. At its start the project was very much driven by efficient transmission of data over networks, which never turned out to be a central problem in any approach to MT

(LdT) We took very little from AI approaches -- as someone said earlier, we did not think the `Bar-Hillel problem' was terribly important: the Bar-Hillel problem is the problem of

ambiguity that can only be resolved on the basis of common sense reasoning, such as in the 'the box is in the pen' problem. Because of this problem, MT was said to be 'AI Complete' and therefore considered by some to be impossible. We thought this view was entirely wrong -- as if a demonstration that there is no ultimate cure for cancer would lead the medical research community to abandon work aimed at developing treatments.

(DA) We assumed it would not be fatal if the system sometimes delivered a number of different linguistically possible translations, some of which quite implausible. But we also hoped to diminish the problem by the use of certain kinds of semantic features in the Interface Structures. I'm sure this is what we thought, but you know, I don't think the issue was ever really discussed in the project, and I think that this is quite revealing: there was often a reluctance in the project to discuss very theoretical issues, in favour of getting on and `doing it'.

(SK) TAUM Meteo and Aviation: we found them very good, we thought of them as friends and a source of inspiration.

(DA) Unification grammar, such as LFG: I think that if these grammar formalisms had existed when the project started, they would have been strong candidates to be adopted as the grammatical formalism. As it was, they came on the scene rather too late for us. Still, the CAT formalism makes extensive use of unification in the way structures were built up. Its an interesting question whether the project would have been more successful it had been more unification based. I'm not sure about this, but while it might have made for an overall cleaner system and a better way of addressing some linguistic problems, it would not have automatically lead to solutions to other fundamental problems.

(SK) Example based translation. The first publication on this was by Nagao in 1984, which was much too late to have any real impact on our thinking, but in any case it would probably not have had much impact because it did not fit in our paradigm, which was really based on grammar.

Was there an isolationist tendency in the project?

(LdT) On the contrary, internally, at least, this was the most open and diverse working society that you can imagine. We were constantly working with people from all the other European countries, with all sorts of different scientific backgrounds. This was very novel for most of us, and also very interesting and exciting. But it was also a problem, the DTSB problem mentioned above -- the 'different scientific background and training' problem.

(SK) And we took quite a lot of ideas from other projects, for example the whole basis of dependency grammar came from GETA, and we were very much influenced by ideas from the Unification grammar approaches.

But one failure was that no one in the project ever carried out a proper review of the state of the art, and this was something we should have done, in retrospect. Such a thing should be part of every scientific endeavour.

Evaluation, Reflection, Legacy

How successful were you at the time, and how successful do you now think you were?

(LdT) At a purely practical level, in terms of the project goals, we did not achieve the development goal of producing a working, usable, practical system. But at the time my main reason for being negative about the project was that I felt that we could have achieved more with the group of people we worked with and the and the resources we had. For example, in our own Dutch group, we assigned a few people to explore a slightly different approach based on a straightforward application of DCGs, which actually lead to a fairly interesting working system, the so-called MiMo system.

(SK) In some ways I'm more positive now than I was then about the project - because I think that other projects of that era did not do much better, e.g. Rosetta, the Japanese projects and DLT. We did not do much worse than the others, and in some ways we did better. And I see that many of the problems are still unsolved.

(DA) One undeniable success was that the project produced a European computational linguistics community -- a relatively large number of well-trained computational linguists, many of them in countries like Spain, Portugal, and Greece, which simply had no tradition of computational linguistics before they joined Eurotra.

It has really led to an internationalization of research in Europe -- where previously most linguists and computational linguists in Europe been working from a rather limited, national perspective, it has become the normal thing to think internationally, to regard colleagues from other countries as genuine colleagues and potential collaborators. Of course, this was part of a general movement, e.g. around this time European Chapter of the ACL, EACL, was set up, but Eurotra

was special in that people had real experience of working together on a practical level.

Which of your methods do you think could still be used today?

(DA) If we take this totally literally, none. If we take it more abstractly ideas like modularity, separation of data and programs, firm linguistic foundations, a principled approach to transfer, limited interlinguality, are still viable and important ideas.

The interviewees

Steven Krauwer
has a degree in mathematics and has worked as a lecturer and researcher in computational linguistics and language technology at the Utrecht institute of Linguistics (UiL OTS) of Utrecht University since 1972. He has been the coordinator of a number of EU-funded projects in the field of language and speech technology. He has recently retired but is still working for UiL OTS as an affiliate researcher and project manager, including being the coordinator of Elsnet.

Louis des Tombe
trained as a psychologist and taught computational linguistics and psycholinguistics at the Faculty of Arts of Utrecht University from 1970 until his retirement in 2005. After the end of the Eurotra project he worked as a member of the faculty ICT team.

Doug Arnold
studied English Literature as an undergraduate, and has taught computational and theoretical linguistics at the University of Essex since 1981. Since the end of the Eurotra project he has continued teaching and research in computational and theoretical linguistics.

Reading and Further Information

The project ended long before the invention of the World Wide Web, but some information, and more links to other sources of information about the project can be found at EUROTRA 1978 - 1992 (http://www-sk.let.uu.nl/eurotra/index.html). A photo of the participants at the Bangor workshop can be found at http://www-sk.let.uu.nl/eurotra/bangor.html.

Early discussion of Eurotra appears in [King and Perschke(1982)], [Johnson et La.(1985)Johnson, King, and des Tombe], and [King and Perschke(1987)]. Later work is discussed in [Copeland et al. (1991a)], and [Copeland et al. (1991b)], and in two special issues of the Journal *Machine Translation* (Vol. 6, numbers 2 and 3, 1991). The CAT approach is described in [Arnold and des Tombe(1987)] [Arnold et al.(1985)Arnold, Jaspaert, Johnson, Krauwer, Rosner, des Tombe, Varile, and Warwick] one development of it, called Mimo is described in [Arnold and Sadler(1990)], and a development of that called Mimo2 is described in [Noord et al.(1991)Noord, Dorrepaal, Eijk, Florenza, Ruessink, and Tombe].

The ALPAC report is [Pierce and Carroll(1966)]. Systran is described in [Wheeler(1987)], [Bar-Hillel(1951)] first described the `Bar-Hillel problem'. The TAUM system is described in [Isabelle(1987)], [Isabelle and Bourbeau(1985)]; the GETA system is described in [Boitet(1987)]. The Rosetta system is described in [Rosetta(1994)]. References on the Japanese national project include [Nagao et al.(1988)Nagao, Tsujii, and Nakamura], Nagao's idea about Example Based Translation is set out in [Nagao(1986)]. The DLT system is described in [Maxwell et al.(1988)Maxwell, Schubert, and Witkam]. The METAL system is described in [Fontenelle et al.(1994)Fontenelle, Adriaens, and Braekeleer], [Slocum(1987)], [Gebruers(1988)] and [Bennett and Slocum(1988)].

LFG is described in [Kaplan and Bresnan(1982)].

References

[Arnold and des Tombe(1987)]
D.J. Arnold and L. des Tombe. Basic theory and methodology in EUROTRA. In S. Nirenburg, editor, *Machine Translation: Theoretical and Methodological Issues*, pages 114-135. CUP, Cambridge, 1987.

[Arnold et al.(1985)Arnold, Jaspaert, Johnson, Krauwer, Rosner, des Tombe, Varile, and Warwick]
D.J. Arnold, L. Jaspaert, R.L. Johnson, S. Krauwer, M. Rosner, L. des Tombe, G.B. Varile, and S. Warwick. A Mu-1 view of the <<C,A>,T> framework in EUROTRA. In *Proceedings of the First International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 1-14, Colgate University, New York, 1985.

[Arnold and Sadler(1990)]
Doug Arnold and Louisa Sadler. The theoretical basis of MiMo. *Machine Translation*, 5(3):195-222, 1990.

[Bar-Hillel(1951)]

Y. Bar-Hillel. The state of translation in 1951. *American Documentation*, 2:229-237, 1951. (Reprinted in Y. Bar-Hillel, *Language and Information*, Addison Wesley, Reading, Mass., 1964, pages 153-165).

[Bennett and Slocum(1988)]
W. Scott Bennett and J. Slocum. METAL: the LRC machine translation system. In J. Slocum, editor, *Machine Translation systems*, pages 111-140. CUP, Cambridge, 1988.

[Boitet(1987)]
C. Boitet. Research and development on MT and related techniques at Grenoble University (GETA). In M. King, editor, *Machine Translation Today*, pages 133-153. Edinburgh University Press, Edinburgh, 1987.

[Copeland et al. (1991a)]
C. Copeland, J. Durand, S. Krauwer, and B. Maegaard, editors, The Eurotra Linguistic Specifications, CEC, Luxembourg, 1991.

[Copeland et al. (1991b)]
C. Copeland, J. Durand, S. Krauwer, and B. Maegaard, editors, The Eurotra Formal Specifications, CEC, Luxembourg, 1991.

[Fontenelle et al.(1994)Fontenelle, Adriaens, and Braekeleer]
Thierry Fontenelle, Geert Adriaens, and Gert De Braekeleer. The lexical unit in the METAL MT system. *Machine Translation*, 9(1):1-19, 1994.

[Gebruers(1988)]
Rudi Gebruers. Valency and MT: Recent developments in the METAL system. In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 168-175, Austin-Marriott at the Capitol, Austin, Texas, USA, 1988. URL http://www.aclweb.org/anthology/A88-1023.pdf.

[Isabelle(1987)]
P. Isabelle. Machine Translation at the TAUM Group. In M. King, editor, *Machine Translation Today*, pages 247-277. Edinburgh University Press, Edinburgh, 1987.

[Isabelle and Bourbeau(1985)]
Pierre Isabelle and Laurent Bourbeau. TAUM-Aviation: Its technical features and some experimental results. *Computational Linguistics*, 11(1):18-27, 1985. URL http://www.aclweb.org/anthology/J85-1002.pdf.

[Johnson et al.(1985)Johnson, King, and des Tombe]
Rod Johnson, Maghi King, and Louis des Tombe. EUROTRA: A multilingual system under development. *Computational Linguistics*, 11(2-3): 155-169, 1985. URL http://www.aclweb.org/anthology/J85-2005.pdf.

[Kaplan and Bresnan(1982)]
R.M. Kaplan and J. Bresnan. Lexical Functional Grammar: a formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173-282. MIT Press, Cambridge, Mass, 1982.

[King and Perschke(1982)]
M. King and S. Perschke. EUROTRA and its objectives. *Multilingua*, 1:27-32, 1982.

[King and Perschke(1987)]

M. King and S. Perschke. EUROTRA. In M. King, editor, *Machine Translation Today*, pages 373-391. Edinburgh University Press, Edinburgh, 1987.

[Maxwell et al.(1988)Maxwell, Schubert, and Witkam]
D. Maxwell, K. Schubert, and T. Witkam, editors. *New Directions in Machine Translation*. Foris Pub., Dordrecht, 1988.

[Nagao(1986)]
M. Nagao. *Machine Translation -- How Far Can It Go?* Oxford University Press, Oxford, 1986.

[Nagao et al.(1988)Nagao, Tsujii, and Nakamura]
M. Nagao, J. Tsujii, and J. Nakamura. The Japanese Government Project for Machine Translation. In J. Slocum, editor, *Machine Translation Systems*, pages 141-186. CUP, Cambridge, 1988.

[Noord et al.(1991)Noord, Dorrepaal, Eijk, Florenza, Ruessink, and Tombe]
Gertjan Van Noord, Joke Dorrepaal, Pim Van Der Eijk, Maria Florenza, Herbert Ruessink, and Louis Des Tombe. An overview of MiMo2. *Machine Translation*, 6 (3): 201-214, 1991.

[Pierce and Carroll(1966)]
John R. Pierce and John B. Carroll. *Language and Machines -- Computers in Translation and Linguistics (ALPAC Report)*. National Academy of Sciences, Washington D.C., 1966.

[Rosetta(1994)]
M.T. Rosetta. *Compositional Translation*. Kluwer Academic Publishers, Dordrecht, 1994.

[Slocum(1987)]
J. Slocum. METAL: the LRC machine translation system. In M. King, editor, *Machine Translation Today*, pages 319-350. Edinburgh University Press, Edinburgh, 1987.

[Wheeler(1987)]
P. Wheeler. SYSTRAN. In M. King, editor, *Machine Translation Today*, pages 192-208. Edinburgh University Press, Edinburgh, 1987.