MT Express

Rémi Zajac
Computing Research Laboratory, New Mexico State University
zajac@crl.nmsu.edu

We describe a Machine Translation framework aimed at the rapid development of large scale robust machine translation systems for assimilation purposes, where the target MT system is incorporated as one of the tools in an analyst's workstation.

1 Introduction

The machine translation systems that are being developed at CRL are designed for assimilation purposes and are targeted at a large variety of source texts, including news articles, Web pages, newsgroups articles and email traffic. Thus, coverage and robustness are emphasized over depth of analysis, and accuracy over stylistic fluidity. Moreover, these systems are for the most part developed under severe resource constraints. Some of the new languages which are or will be covered are so-called 'low-density languages': languages for which there are little or no electronic resources, comparatively little expertise and few descriptive linguistic works published. An example of such a language under development at CRL is Persian. The lack of electronic resources, including bilingual corpora or even monolingual corpora rules out statistical and learning-based approaches to machine translation. As a consequence, language resources are carefully structured and the organized to support rapid and large scale acquisition of resources (computational dictionaries and grammars). Robustness is also a fundamental issue, and the architecture of the machine translation system itself is designed to produce translations even with incomplete resources (although breadth of lexical coverage is a minimum requirement). One of the desiderata of the MT design is the ability to produce translations after a very short period of development: the incremental addition of linguistic knowledge in the system improves the translation quality without the need to restructure already acquired knowledge. In this paper, we illustrate how the chosen structuration of the language resources supports on the one hand rapid and incremental acquisition of resources and enables robust processing on the other hand.

1.1 Past experience: the Temple project

One of the results of the Temple project at CRL, a three year effort in building a set of MT systems translating from Arabic, Japanese, Russian and Spanish to English with low amount of resources (Vanni & Zajac 97), is that a carefully designed MT architecture is crucial for developing MT systems with a minimal amount of effort, and that the quality of the software contributes significantly to the quality of the final result. The various Temple MT systems were built reusing existing components and resources whenever they existed, even if the quality was low. This experience taught us some important lessons on the construction of robust machine translation systems. In particular, it is very difficult to avoid error compounding and to make sure that the final actual quality of the translation is as good as the quality of the weakest component of the system. Also, various levels of linguistic analysis were identified and their relationship made precise not only for the purpose of robust multilevel processing, but also for minimizing the effort in acquiring and maintaining the linguistic resources used by the various components, and ensuring a uniform quality across all these resources. Finally, since these machine translation system were developed with levels of funding and resources which varied over time, the issue of scalability rose to prominence, and is related to both the multilevel linguistic approach and to the architecture of the MT system software itself.

At the end of the Temple project, we started a new effort, the Corelli project, for building an integrated machine translation architecture that would fully meet these requirements. This new MT architecture is also one of the target of the new Expedition project at CRL (Nirenburg & Raskin 98), which aims at building an integrated linguistic knowledge elicitation environment to develop languages resources for building a machine translation system in a very short period of time, with a limited number of human resources, and for any low-density language.[1] Since one of the constraints is that the human acquirers are not linguists or computational linguists, and have no prior knowledge of machine translation, or even natural language processing at all, any knowledge about the processing and the control flow in the system should be hidden; the acquirers should not need to specify any kind of procedural knowledge. One feature of the Corelli architecture is precisely that all linguistic knowledge is expressed in a declarative way to a large extent. Some procedural information still needs to be specified, but at a fairly high level (e.g., that morphological analysis is applied after tokenization in some languages, and that the syntactic parser is applied after morphological disambiguation).

1.2 Goals: coverage, robustness and incremental development

The multilevel linguistic representation used in the architecture is motivated by two sets of goals. The first set of goals is pragmatic. One goal is facilitating the *construction* of a syntactic model and the *acquisition* of syntactic rules as well as syntactic zones in a dictionary. In particular, the acquisition of lexical entries is deemed one of the most expensive tasks in the process of building an NLP system and special attention is paid to reduce this acquisition effort as much as possible. A second goal is enhancing robustness of the various processors. Although an important part of the robustness factor is tied to the kind of processing itself, it is largely constrained by the way linguistic information is structured: we strive at defining a modular framework where each syntactic module has a few well-defined interactions a small number of other modules (ideally, only one or two others). Failure of one module should have minimal consequences on the overall output quality of the system.

The second set of goals is related to the targeted applications, that is machine translation systems. The way of encoding syntactic information should facilitate the construction of bilingual transfer dictionaries as well as syntactic transfer rules. In particular, an incremental and modular approach to the development of language resources is deemed essential: the construction of a machine translation system is very complex and it is realistically impossible to wait until the completion of all modules at the expected depth of analysis. A staged and modular approach has two important consequences:

- It becomes possible to test the system throughput on actual documents very early in the development cycle;
- Each module can be tested and debugged independently of others without waiting for the completion of the whole system (testing a complete system without being able to test each module independently is a nightmare that any MT developer dreads).
- And last but not least, it becomes possible to convince funders early in the project that the project's money will not be wasted in some new hopeless MT venture.

This paper presents the Corelli architecture and shows how it addresses the challenges enumerated above. Section 2 presents the robust scalable parsing framework which enables translation at varying depths of linguistic representation depending on the availability of the corresponding linguistic knowledge in the dictionaries and the parser's rules. Section 3 gives an overview of the multilevel linguistic representation used in the system and shows that it addresses the needs for robustness and scalability as well as the need to facilitate acquisition and maintenance of linguistic resources. This representation provides a standardized

[1] Project requirements mention a transfer-based MT system developed from scratch by a team of one language specialist (e.g., a translator) and one programmer in 6 months; the English generation and the English target dictionary, as well as the MT engines are provided and the team has to build language resources for analysis and transfer only (!).

framework for linguistic description that can be applied to a large variety of languages. Section 4 presents briefly the incremental acquisition strategy followed in developing languages resources for a machine translation system.

2 Robust Machine Translation

Robust machine translation can be achieved by a combination of:

- Breath of lexical coverage;
- Robustness of each individual component (e.g., of the morphological analyzer, which must include a full grammar of unknown words and recognize genuine unknown words from proper names or misspellings);
- Flexible organization of the set of components to provide fall-back in case of failure of one of the components.

The Corelli MT architecture offers the functionalities necessary to implement a robust top-level organization, and specialized rule formalisms are also designed with robustness as a requirement.

2.1 Process

We divide syntactic analysis into the following steps:

1. Morphological disambiguation based on the definition of allowed and forbidden patterns (sequences of words). Although this is not strictly speaking syntax, some of the information that is used in the disambiguation patterns is of syntactic nature, and the format of the disambiguation patterns is similar to the one for syntactic rules.
2. Recognition of phrasal boundaries, done in two sub-steps: (1) determination of the placement of left or right phrasal boundaries, and (2) matching boundaries to form parenthetical structures. Transfer rules at this level map source constituents to target constituent, doing no more than basic reordering of constituents (see e.g., Furuse & Iida 96).
3. Construction of dependency structures using dependency rules. Each rule is applied within phrasal boundaries: avoiding matching dependency rules across boundaries speeds up the analysis process. Transfer rules at this level map also define reordering among heads and dependents, but may also lexicalize information carried as the value of some features in heads (e.g., negation, modality, etc.).
4. Disambiguation of dependency structures: subcategorization information is used to rule out dependency structures where arguments are not correctly attached to the head.
5. Construction of argument structure: subcategorization is used to assign constituents as arguments of heads. This step is done at the same time as the previous step.

At this level of analysis, we usually have complete simple sentence descriptions, inclusion of sentence-level prepositional phrases, clauses, subordinations and conjunctions, and phrases. One level is still missing: the analysis of complex sentential patterns, such as ‘either X or Y’. Although some complex sentential structures might be analyzed using some discourse or text structure theory, we do not envisage that full coverage will be achieved. Therefore, the last step will be the application of a grammar to analyze this kind of pattern. At this level again, transfer rules will be define to translate each of these patterns.

Note that transfer rules are defined for each level of linguistic representation and can therefore be applied after each analysis step starting from morphology (transfer after morphology is a simple word-for-word translation enhanced with transfer of morphosyntactic features). All parsing is done essentially bottom-up and the input is processed level after level in a sequential fashion. Various levels of analysis can be achieved in a single sentence, and for any input segment, the highest level of transfer is chosen. For example, the parser might fail to analyze a whole sentence but succeed in analyzing the clause structures. In this case, clauses will be translated at the level attained by the parser for each clause, and other sentence elements will be translated word-for-word.

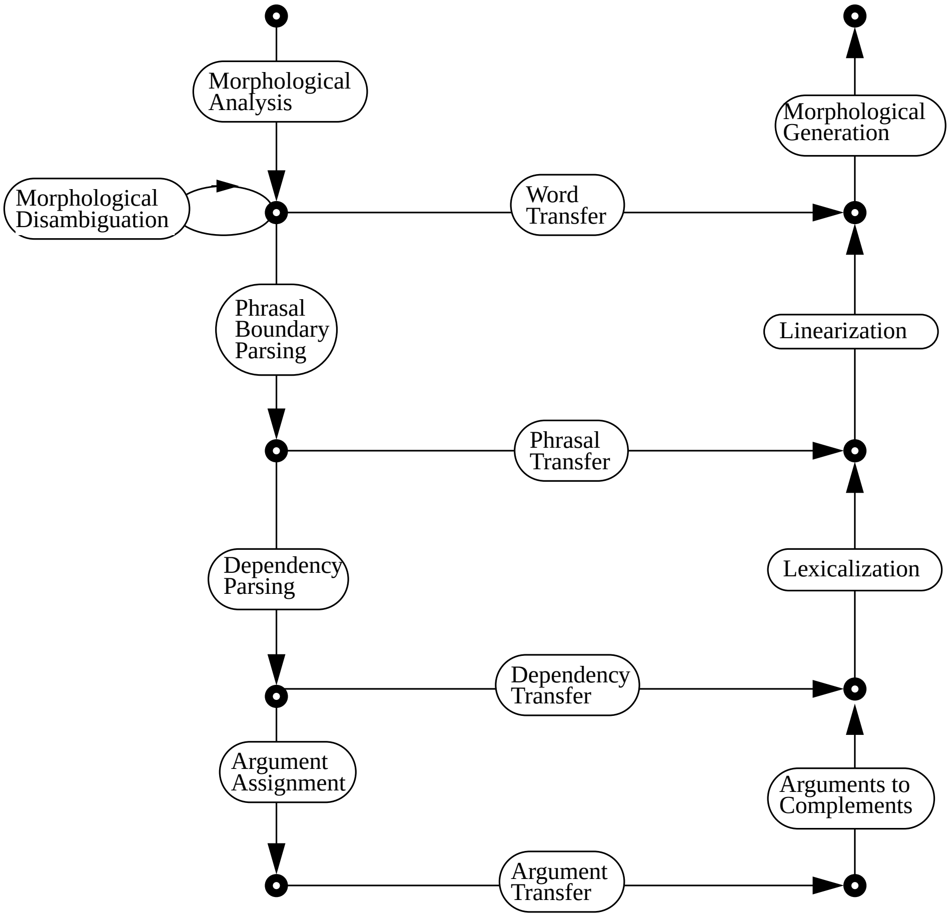

Figure 1: Multilevel Machine Translation.

2.2 Architecture

The Corelli machine translation architecture supports both the development phase and the runtime system. The development version is designed to support interactive acquisition and modification of language resources as well as testing and debugging a whole MT system. A Corelli machine translation system contains a set of linguistic components: the top-level of the system is a graph which defines the control flow between different components. This architecture uses directly the Corelli Document Manager (Zajac et al.

97) which provides an infrastructure and tools for integrating NLP components to build NLP systems (Figure 2).

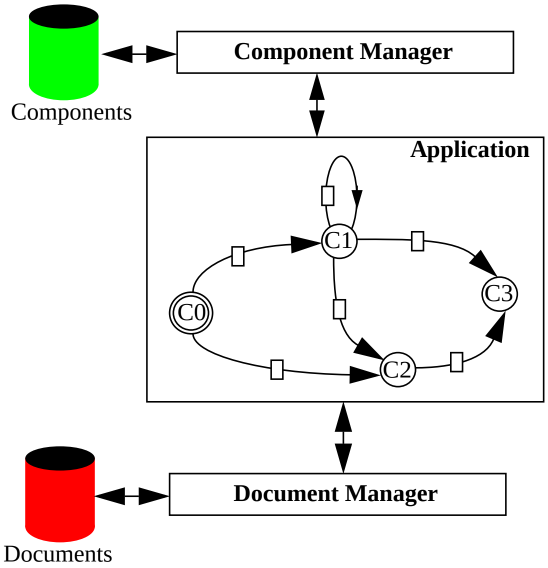

Figure 2: Corelli MT Software Architecture.

The various software components that are used by computational linguists to build an MT system are:

- Tango, a language for defining typed feature structures which provides a set of predefined types (including regular expressions) and supports the notion of modules (package).
- Habanera, a Lexical Knowledge Base management system which is used for managing all lexical resources (Zajac 97b); Lexical entries are instances of typed feature structures and a dictionary schema is defined by type definitions in a Tango module. Habanera supports several indexing schemes which allow runtime access by various NLP engines.
- Samba, a morphological formalism which provides a high-level language for specifying morphological models. Morphological rules map string expressions to feature structures and Samba provides constructions to combine and factorize morphological rules in various ways. This formalism supports reversible morphological analysis and generation (Zajac 97a) and is used to implement morphological analyzers and generators.
- Rumba, a syntactic formalism where a grammar is a set of general rewrite rules for analysis (and generation) based on the composition of generalized finite-state transducers. Several Rumba grammars can be applied sequentially on the graph representing an analysis allowing for finer control of grammar application and modularity in grammar development.
- Mambo, a transfer formalism based on (Zajac 89) and (Amtrup 95) that is used to write all transfer components of a machine translation system.

The control flow between morphological, syntactic and transfer components is defined by a control graph similar to a finite-state graph where transitions define conditions and nodes contain executable components. Conditions can state for example that if subcategorization information has not been used to compute argument structure, transfer must use default argument mapping instead of the standard mapping defined in the dictionaries.

3 Multilevel structuring of language resources

The idea of multilevel structuring of linguistic representations can be traced at least as far as (Lamb 66), and has been developed by linguists such as Mel'çuk (see Mel'çuk 88 for a recent presentation). These ideas have been implemented in text generators at Montréal (Kittredge & Polguère 91) for example, and in the context of machine translation, at Grenoble (Vauquois & Chappuy 85), where the multilevel representation is also used to define levels of fall-back in processing in case of failure at higher levels.[1] Thus, multilevel representations have been used chiefly to structure and partition the linguistic knowledge into manageable parts (Emele et al. 92). Our proposal is cogent with previous multilevel approaches but its main goals are essentially pragmatic: to provide a framework for robust NLP and for incremental acquisition of linguistic knowledge. These goals in turn directly influence the definition of levels and the interaction between levels.

3.1 Multilevel grammars

In a machine translation system, syntactic information is distributed and used in various components: (bilingual) dictionaries, syntactic grammars and transfer grammars. It should be possible to check that syntactic information distributed in all these components is coherent, something which has been traditionally difficult to achieve. To facilitate the control of coherence between these components, the linguist formally defines the syntactic structures and the syntactic categories, features and values, which are used in all these components. These definitions take the form of a set of typed feature structures definitions, and these definitions are used by syntactic and semantic checkers to check lexical entries and rules (Zajac 92a, 92b).

For example, syntactic grammar rules will use part-of-speech information encoded in lexical entries to build dependency structures, and subcategorization to build argument structures. Lexical transfer rules map argument structures from a source language to a target language. Structural transfer rules map dependency structures. Thus, for a machine translation system to work correctly, it is essential to ensure that all syntactic information distributed among these components is coherent. The linguist has to define and acquire the following kinds of syntactic information:

- Morphological disambiguation rules;
- Syntactic categories (parts-of-speech);
- Argument structure and subcategorization;
- Constituent boundary recognition;
- Dependency structures;
- Transfer of constituent and dependency structures;
- Transfer of argument structures.

Each grammar performs a well defined simple task which uses only a small part of the information encoded in lexical items. We can distinguish 2 kinds of grammars: disambiguation grammars and structure-building grammars. There are currently only two kinds of disambiguation grammars: morphological disambiguation grammars which eliminate some morphological ambiguities by considering local context, and constituent disambiguation grammars which eliminate constituent structures where the structure of complements of argument taking words does not correspond to the argument structure of the word.

Structure-building grammars are syntactic analysis grammars and transfer grammars. Analysis grammars are:

[1] Although to my knowledge, the fall-back mechanism has never been implemented to its fullest extent.

- A grammar which assigns constituent-boundaries;
- A grammar building the parenthetical structure using the boundaries;
- A grammar building dependencies structures. This grammar uses constituent boundaries to speed-up parsing.

Transfer grammars are:

- Lexical transfer grammar derived from the specification of argument structure mapping in the lexicon;
- Morphological transfer grammar which maps morphological features on lexical heads;
- Phrasal transfer grammar which maps source phrase ordering to the target phrase ordering;
- Dependency transfer grammar. This grammar has two sub-grammars: one which maps specifiers, modifiers and adjuncts. i.e. all dependents which are not arguments of the head, and a sub-grammar which defines a default mapping for complements if information about transfer of arguments is missing in the dictionary.

Each grammar at a higher level uses information derived by a grammar at a lower level, but a lower-level grammar cannot ‘see’ information built at higher level. This organization allows incremental development and testing of the MT system. A complete throughput can be obtained as soon as the word-for-word level is reached which necessitate only the following components:

1. A bilingual dictionary containing morphological information and translations;
2. A morphological analyzer;
3. A morphological transfer grammar;
4. A target morphological generator.

Any further development will incrementally build on this basis. Furthermore, even if some component cannot process a whole sentence but only fragments of it, each fragment will be translated using the highest level achieved for this fragment.

3.2 Multilevel information in the dictionary

A dictionary entry (corresponding to a single word-sense) records only four kinds of information:

1. Parts-of-speech (POS),
2. Subcategorization (subcat),
3. Mapping (translation) to a target word-sense,
4. Mapping of source argument structure to the argument structure of the target word-sense.

The part-of-speech information is used by the syntactic parser to *build* the syntactic dependency trees to the exclusion of any other information, including subcategorization. Thus, the POS must encode all information about the range of syntactic dependents of the head of a constituent.

Subcategorization encodes the valency of a complement-taking lexical item, information about the number and position of syntactic arguments (or complements) of a head, and the syntactic type of these arguments. Subcategorization is used by the parser (1) to *disambiguate* between several parse trees by selecting a subset of trees where the attachment of complements is consistent with the subcategorization patterns of the head, and (2) to *assign* subcategorized complements to named arguments of the head.

The strict separation between the 2 kinds of information makes it possible to build a system where subcategorization is missing: if subcategorization is missing, the parser will produce many more ambiguous parse trees, and transfer of arguments will be done using default rules.

Some languages may have a more complex morphology and the dictionary may also contain additional morphological properties, such as the inflectional paradigm of a lexical unit and additional stems. Similarly, in order to map an argument structure to syntactic complements in a given syntactic context, the dictionary may contain the specification of the range of syntactic structures in which a given lexical unit can appear (e.g., that a verb cannot appear in a passive construction).

4 An incremental approach to resource acquisition

Given the cost of building language resources for a machine translation system (the dictionary alone can cost as much as 60% of the total cost of a MT system), one of the most important goal is to minimize the cognitive load for the acquisition of language resources. This implies that acquisition follows a predefined scenario, makes use of high quality but simple tools that include training support and on-line help, and that each step addresses only one simple well-defined task.

The linguist will first define the set of features and values that will be used in all components of the system (by defining types for feature structures). Once this step is done and documented, the type definitions will drive some of the acquisition tools. The linguist will either instantiate parameters for these tools or ask for new specialized tools. The main concern will be to carefully define each acquisition task and prepare a set of training materials and documentation for each task. We give an overview of the two main acquisition tasks, the bilingual dictionary and the grammars.

4.1 Lexical acquisition

Given the robust approach to parsing described above, we can organize the dictionary acquisition tasks in distinct steps, the completion of the first allowing the production of word-for-word translations, and the completion of the each of the following steps providing incremental improvements in the quality of translation. We assume that we start the dictionary acquisition with a list of head words.

Step 1: Morphology and target equivalents

The first step includes:

- The definition of the part-of-speech (and in some languages, additional morphological information such as inflectional paradigms and/or additional stems).
- The identification of the word senses (which are not defined by themselves, only by their translation to a set of equivalents).
- For each word-sense, the list of equivalents (words) in the target language.

At this point, it is already possible to run a morphological analyzer and produce a word-for-word translation. If a syntactic parser is available, a parse tree can be produced and transfer rules applied to the parse tree for reordering of the constituents.

Step 2: Syntactic information

The second step includes the definition of subcategorization for complement-taking lexical items, which can then be used by the parser to eliminate spurious parses, for example for prepositional attachment.

Step 3: Argument structure and selection of target word-senses

The last step includes the mapping to target word-senses (instead of simply words) and the mapping of arguments to the target word argument structure. Mismatches are handled during this acquisition step.

The acquisition of lexical entries at CRL actually follows this approach with several important benefits:

- For each acquisition sub-task, the acquirer uses a simple specialized acquisition tool which is not only simple to build but also simple to use.
- Since the acquirer is less distracted by a complex Graphical User Interface (GUI), he can concentrate better on the task at hand.
- Since the task itself is simple and repetitive, the cognitive load is reduced: the acquirer does not have to switch between different complex procedures, and can thus work faster and with less errors.

4.2 Building analysis and transfer grammars

The development of grammars parallels the steps followed by the machine translation process. Once all features and values for all components are defined, each of the following grammars is developed and tested in turn. The development of these grammars also parallels the development of the lexicon: grammars 1 to 5 use only POS information in the dictionary (and possibly additional morphological lexical properties for the morphological analyzer). Grammar 6 uses additionally subcategorization. Grammar 7 uses the lexical mapping of argument structures.

1. Morphological grammar, morphological transfer grammar.[1]
2. Morphological disambiguation grammar; This grammar will contribute to the improvement of the translation quality by eliminating spurious morphological ambiguity without the need for a full syntactic parser.
3. Phrase boundary recognition, phrase transfer grammar. One of main concerns here will be to make sure that the correct phrase structure is always present in the set of phrasal structures since the phrase boundaries will be used as a heuristics to speed-up the dependency parser. These grammars use only the POS information in the dictionary.
4. Fixed sentence patterns and transfer of these patterns: constructions like ‘either…or…’ are handled by a special lexicalized grammar instead of being defined in the lexicon. This allows the production of acceptable translations with a dictionary containing less information, and without relying on a complex syntactic parser.
5. Dependency grammar; dependency transfer. This grammar builds a dependency tree using the phrase structures computed at the previous step and assigns syntactic functions to dependents of the head. The corresponding transfer grammar maps syntactic functions to build the target dependency structure. This transfer grammar may for example introduce new lexical heads in the target structure.
6. Argument assignment: this grammar uses the syntactic type of the construction (e.g., active vs. passive vs. reflexive, etc.) to map syntactic complements to abstract arguments. If during parsing this assignment fails, the corresponding syntactic structure is eliminated from the result (unless all fail, in which case they are all preserved but marked), improving the quality of the translation.
7. Argument transfer: this grammar is directly derived from the bilingual lexicon.

[1] A complete list here should also include rules for unknown words, dates, proper names, acronyms, etc.

5 Conclusions

We have described a new machine translation architecture aimed at fast development of machine translation systems for assimilation purposes where breadth of coverage and the production of a functional system early in the project are of paramount importance. This architecture is used in several machine translation projects at CRL:

- In the Corelli project itself, for Korean and Serbo-Croatian;
- In the Shiraz project, for Persian;
- In the Expedition project, for Turkish and two other 'low-density' languages;
- In the MINDS project, for porting the Temple Spanish, Japanese and Russian system to the new architecture.

Although this architecture is still under development at the time of writing, several major components have already been implemented, allowing to proceed with the development of the dictionaries and the morphology.

6 References

Amtrup, Jan. 1995. "Chart-based Incremental Transfer in Machine Translation". *TMI'95 – Proceedings of the 6th Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium. pp188-195.

Emele, Martin, Ulrich heid, Stefan Momma and Rémi Zajac. 1992. "Interaction between Linguistic Constraints: Procedural vs. Declarative Approaches". *Machine Translation* 7/1-2, Special Issue on Text Generation. pp61-98.

Furuse, Osamu, Hitoshi Iida. 1996. "Incremental Translation Utilizing Constituent Boundary Patterns". *COLING'96 – Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 5-9 August 1996. pp412-417.

Kittredge, R. I. & A. Polguère. 1991. "Dependency Grammars for Bilingual Text Generation: Inside FoG's Stratificational Models". *Proceedings of the International Conference on Current Issues in Computational Linguistics*, Penang, Malaysia. pp318-330.

Lamb, S. 1966. *Outline of Stratificational Grammar.* Georgetown University Press, Washington D.C.

Igor A. Mel'çuk. 1988. *Dependency syntax: Theory and Practice*. State University Press of New York, Albany.

Nirenburg, Sergei, Victor Raskin. 1998. "Universal Grammar and Lexis for Quick Ramp-Up of MT Systems". Submitted for COLING'98.

Michelle Vanni and Rémi Zajac. 1997. "Glossary-Based MT Engines in a Multilingual Analyst's Workstation Architecture". *Machine Translation* 12, Special Issue on New Tools for Human Translators. pp131-157.

Bernard Vauquois and Sylviane Chappuy. 1985. "Static Grammars". *Conference on Theoretical and Methodological Issues in Machine Translation*, Colgate University, 14-16 August 1985.

Rémi Zajac. 1997a. “Feature Structures, Unification and Finite-State Transducers”. Submitted for *FSMNLP’98, International Workshop on Finite State Methods in Natural Language Processing*, June 29 - July 1, 1998. Bilkent University, Ankara, Turkey.

Rémi Zajac. 1997b. “Habanera – A Multipurpose Multilingual Lexical Knowledge Base”. Workshop on Multilingual Natural Language Processing, *NLPRS’97, Natural Language Processing Pacific Rim Symposium*, 2-4 December 1997, Phuket, Thailand.

Rémi Zajac, Mark Casper and Nigel Sharples. 1997. “An Open Distributed Architecture for Reuse and Integration of Heterogeneous NLP Components”. *ANLP’97, 5th Applied Natural Language Processing Conference*, 31 March - 3 April 1997, Washington D.C. pp245-256.

Rémi Zajac. 1992a. “Inheritance and Constraint-based Grammar Formalisms”. *Computational Linguistics* 18/2, June 1992, pp159-182.

Rémi Zajac. 1992b. “Towards Computer-Aided Linguistic Engineering”. *COLING’92 – Proceedings of the 14th International Conference on Computational Linguistics, 23-28 August 1992, Nantes, France. pp828-834.*