INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Atoll

Atelier d'Outils Logiciels pour le Langage naturel

Rocquencourt

THEME SYM

Table of contents

1. Team

Head of project team
Éric Villemonte de la Clergerie [CR]

Vice-head of project team
Pierre Boullier [DR, HdR]

Administrative assistant
Nadia Mesrar [TR]

Staff members Inria
Bernard Lang [DR]
Philippe Deschamp [CR]
François Thomasset [DR]

External members
François Barthélemy [Maître de conférences, CNAM]
Areski Nait Abdallah [Professeur, Univ. of Brest]
Alexis Nasr [Professor, TALANA, Univ. of Paris 7, delegation until June 2006]

Visiting scientists
Francisco José Ribadas Pena [3 months, Nov. 2005 - March 2006, University of Courña]
Vitor Rocio [1 week, December 2006, University of Lisboa]

Ph. D. student
Benoît Sagot [Détachement du corps des Télécoms, until July 2006]

Technical staff
Isabelle Cabrera [since September 2006]

Student intern
Alexandra Mounier [CNAN Engineer Internship, til September 2005]
Ol'ga Feiguina [Internship, University of Montreal, May-June 2006]
Milagro Fernandez Gavilanes [University of Vigo]
Lionel Nicolas [co-supervised master thesis, University of Nice]
Julien Martin [master thesis, University of Paris 6]

2. Overall Objectives

2.1. Tools for Natural Language Processing

Project-team ATOLL was formed by people with strong competences in Parsing, essentially acquired in the context of Programming Language Compilation. This competence is now applied to *Natural Language Processing* (NLP), mainly in its parsing aspects but evolving toward more semantic aspects. Besides promising industrial applications, this domain of research also offers many scientific problems that may benefit from a strong formal and algorithmic approach.

In our exploration of fundamental parsing techniques, we focus on the use of tabular techniques, almost mandatory to efficiently handle the ambiguities inherent in any human language. The genericity of our techniques is also an asset because of the large diversity of grammatical formalisms. We also explore more recent and important issues related to robustness. We validate these techniques through the development of two prototype environments (SYNTAX and DYALOG) that may be used for building and running parsers.

However, a parser is only one component of a linguistic processing chain that requires other tools and also linguistic resources like lexicons. Besides interesting software engineering issues, designing and running such a chain raises questions about the availability and reusability of linguistic resources. These observations motivate our interest about the standardization, distribution and exploitation of linguistic resources. In particular, we explore how the production cost of some linguistic resources could be reduced by using automatic or semi-automatic acquisition methods, possibly based on parsing corpora with our parsers.

Obviously, such an approach is also an opportunity to test ATOLL's tools on a larger scale. We also believe that the use of well-designed tools for linguists can speed up the hand-crafting of linguistic resources, as we try to promote with Meta-Grammars, a level of abstraction above grammars allowing easier linguistic descriptions.

From a wider point of view, the acquisition of linguistic resources share some common aspects with the extraction of information from corpora or documents, a rapidly growing domain of research and applications. Indeed, the huge development of the World Wide Web and the recent emergence of the notion of Semantic WEB plead for accessing information rather than simply accessing raw documents. As a consequence, tools are needed for extracting information from documents.

The diversity of the tools and resources needed to process natural language overcomes the capacities of project-team ATOLL. Therefore, we favor partnerships for reusing existing tools and resources or for developing new ones in common. An important issue, related to these cooperations and also very present in the NLP community, concerns the standardization and reusability of these tools and resources.

While marginal within ATOLL but nevertheless related to better accessing linguistic resources and tools, a reflexion is led by Bernard Lang on the issues of free access to scientific and technical resources, issues whose scientific, economical, and political interest becomes more and more visible.

3. Scientific Foundations

3.1. Grammatical formalisms

Keywords: *NLP*, *Parsing*, *computational linguistics*, *dynamic programming*, *logic programming*.

Participants: Pierre Boullier, Éric Villemonte de la Clergerie, Benoît Sagot.

CFG *Context-Free Grammars*

DCG *Definite Clause Grammars*

TAG *Tree Adjoining Grammars*

TIG *Tree Insertion Grammars*

LIG *Linear Indexed Grammars*

LFG *Lexical Functional Grammars*

HPSG *Head-driven Phrasal Structure Grammars*

RCG *Range Concatenation Grammars*

MCG *Mildly Context-sensitive Grammars*

LPDA *Logical Push-Down Automata*

2SA *2-Stack Automata*

TA *Thread Automata*

Dynamic Programming Algorithmic method based on dividing a problem into elementary sub-problems whose solutions are tabulated to be reused whenever possible

This theme explores the use of generic parsing techniques covering a large continuum of NLP grammatical formalisms, focusing especially on efficient handling of ambiguities.

3.1.1. From programming languages to linguistic grammars

The shift from programming language grammars to NLP grammars seriously increases complexity and requires ways to handle the ambiguities inherent in every human language. It is well known that these ambiguities are the sources of many badly handled combinatorial explosions.

Furthermore, while most programming languages are expressed by (subclasses) of well-understood context-free grammars (CFGs), no grammatical formalism has yet been accepted by the linguistic community for the description of human languages. On the contrary, new formalisms (or variants of older ones) appear constantly. Many of them may be classified into the two following large families:

Mildly context-sensitive formalisms : They manipulate possibly complex elementary structures with enough restrictions to ensure the possibility of parsing with polynomial time complexities. They include, for instance, Tree Adjoining Grammars (TAGs) with trees as elementary structures, Linear Indexed Grammars (LIGs), and Range Concatenation Grammars (RCGs).

Unification-based formalisms : They combine a context-free backbone with logic arguments as decoration on non-terminals. Most famous representatives are Definite Clause Grammars (DCGs) where PROLOG powerful unification is used to compute and propagate these logic arguments. More recent formalisms, like Lexical Functional Grammars (LFGs) [42] and Head-Driven Phrasal Structure Grammars (HPSGs) [48] rely on more expressive Typed Feature Structures (TFS) [40] or constraints.

The above-mentioned characteristics may be combined, for instance by adding logic arguments or constraints to non-terminals in TAGs. We should also mention that we also concur to this large diversity of formalisms with the introduction of RCGs (Section 6.1).

However, despite this diversity, most formalisms take place in a so-called **Horn continuum**, i.e. a set of formalisms with increasing complexities, ranging from Propositional Horn Clauses to first-order Horn Clauses (roughly speaking equivalent to PROLOG), and even beyond.

This observation motivates our exploration of generic parsing techniques covering this continuum, through two complementary approaches. Both of them use dynamic programming ideas to reduce the combinatorial explosions resulting from ambiguities :

Multi-pass approach : Parsing is broken into a sequence (or cascade) of parsing passes, of (practical or theoretical) increasing complexities, each phase guiding the next one ;

Global Approach : It is mainly based on the use of Push-Down Automata [PDA] and extensions to describe parsing strategies for complex formalisms.

These two approaches enrich each other: studying some specificities observed for the multi-pass approach has triggered theoretical advances; conversely, well-understood and identified theoretical concepts have suggested a widening of the scope of the multi-pass approach.

3.1.2. Multi-pass approach

Programming languages processing is usually broken into several successive phases of increasing complexity : lexical analysis, parsing, static semantics,... The decomposition is motivated by theoretical and practical reasons. The finite state automata (FSA) that model lexical analysis are very efficient but do not have enough expressive power to describe the syntax, which requires, at least, Context-Free Grammars. Similarly, CFGs are not powerful enough to describe some contextual phenomena needed in static semantics. Beside a better efficiency (each phase being handled with the best level of complexity), decomposing increases modularity.

The multi-pass approach for NLP results from similar observations. We try to identify and capture, within adequate grammatical formalisms, subparts of grammars which can guide the remaining processing. For instance, we observe that most formalisms found in the Horn continuum are structured by a non-contextual backbone. This backbone may be first parsed with a very efficient and generic non-contextual parser, namely SYNTAX (cf. 5.1). More formalism-specific treatment can then be applied to check additional constraints, as done this year for LFG decorations (cf. 6.1).

3.1.3. Global approach

The multi-pass approach is less easy to implement when there is no obvious decomposition, for instance when the CF backbone of a formalism cannot be extracted (as in PROLOG) or when the possible phases would be mutually dependent (for instance, when some constraints have a strong impact on the processing of the CF backbone). A more global approach is then needed where constraints and parsing are handled simultaneously.

This very general approach relies on abstract Push-Down Automata formalisms that may be used to describe parsing strategies for various unification-based formalisms [16]. The notion of stack allows us to apply dynamic programming techniques to share elementary sub-computations between several contexts : the intuitive idea relies upon temporarily forget information found in stack bottoms. Elementary sub-computations are represented in a compact way by *items*. The introduction of 2-Stack Automata [2SA] allowed us to handle formalisms such as TAGs and LIGs [17], [1]. More recently, *Thread Automata* (TA) [14] have been introduced to cover mildly-context sensitive formalisms such as Multi-Component TAGs (MC-TAGs).

This global approach may be related to *chart parsing* [41] or *parsing as deduction* [47] and generalizes several approaches found in Parsing but also in Logic Programming. The DyALog system (cf. 5.2) implements this approach for Logic Programming and several grammatical formalisms.

3.1.4. Shared parse and derivation forests

Both previously presented approaches share several characteristics, for instance the use of dynamic programming ideas and also the notion of *shared forest*. A shared forest groups in a compact way the whole set of possible parses or derivations for a given sentence. Formally, a shared forest may be seen as a grammar or a logic program [9]. For instance, parsing with a CFG may lead to an exponential (or unbounded) number of parse trees for a given sentence, but the parse forest remains cubic in the length of the sentence and is itself equivalent to a CFG (as an instantiation of the original CFG by intersection with the parsed sentence). Moreover, these shared forests are natural intermediary structures to be exchanged from one pass to the next one in the multi-pass approach. They are also promising candidates for further linguistic processing (semantic processing, translation, ...). One can also relatively easily extract dependency information between words from these forests, as done in the context of the parsing evaluation campaign EASy. Disambiguation algorithms can also be applied on such shared structures.

3.2. Linguistic Infrastructure and Standardization

Participants: Éric Villemonte de la Clergerie, Benoît Sagot, Pierre Boullier, Philippe Deschamp, François Thomasset.

We are interested in the many issues related to the installation of a whole linguistic processing chain, in particular for accessing and representing the needed linguistic resources and for processing raw texts before sending them to our parsers (cf. 6.8).

To facilitate the installation of such linguistic chains, we develop two systems to build parsers, namely SYNTAX (cf. 5.1) and DyALog (cf. 5.2). We also develop and distribute several linguistic components (cf. 5.4).

Because we realized that diffusing or reusing tools and resources is not really possible without some standardization, ATOLL is involved in national and international efforts to standardize linguistic resources, in particular using XML-based representations. This decision follows preliminary experimentations we have conducted to standardize TAGs and shared forests.

3.3. Resource acquisition and crafting

Participants: Éric Villemonte de la Clergerie, Benoît Sagot.

MG *Meta-Grammars*

Linguistic resources are scarce and expensive, because they are difficult to build, especially when hand-crafted. This observation motivates us to investigate methods to automatically or semi-automatically acquire, supplement and correct linguistic resources. Successful experiments have been conduced with different languages for the automatic acquisition of morphological knowledge from raw corpora. We would like to investigate also a higer bootstrap level where parsing corpora may be used to enrich lexica that may themselves be used for better parsing.

Preliminary experiments have been conducted during the now ended ARC (Action de Recherche Concertée) RLT « Linguistic resources for TAGs » and we are currently working on processing botanical corpora (cf. 6.10).

For hand-crafted resources, we try to design adequate tools and adequate levels of representation for linguists. For instance, we are currently involved in developing grammars through a more abstract notion of *Meta-Grammar* (MG) (cf. 6.6). Introduced by [39], a Meta-Grammar allows the linguist to focus on a modular description of the linguistic aspects of a grammar, rather than focusing on the specific aspects of a given grammatical formalism. Translation from MGs to grammatical formalisms such as TAG or LFG may be automatically handled (cf. 5.3). Graphical environments can be used to design MGs and their modularity provides a promising way for sharing the description of common linguistic phenomena across human languages [8].

The constitution of resources such as lexica or grammars raises the issues of the evaluation of these resources to assess their quality and coverage. For this reason, ATOLL has been deeply involved during 2004 in the Parsing Evaluation campaign EASY and has continued working on these issues in 2006 (cf. 7.4). We have also started investigating different kinds of feedback mechanisms to detect problems when using resources (unknown words, error mining, ...), cf. 6.9.

4. Application Domains

4.1. Applications

Computational Linguistics offers a wide range of potential applications, especially with the emerging of information systems. More specifically for ATOLL, one can (non exhaustively) list the following application domains:

Grammatical checking Parsing is used to detect grammatical errors and to suggest corrections. Tabulation-based parsing techniques present a great potential for grammatical checking because they allow the exploration of many alternatives (for correcting errors) without combinatorial explosions.

Knowledge acquisition Linguistic (and statistical) techniques may be used to extract knowledge from corpora, ranging from a simple terminological list of words to more complex semantic networks with concepts and relations. In this continuum, we also find lexicons, thesaurus, and ontologies. We strongly believe that this domain can benefit from more sophisticated parsing-based techniques.

Text mining and Questions/Answers Parsing and possibly semantic or pragmatic processing may be used to extract precise information from a document, for instance to feed a (knowledge) database or to answer questions formulated by users.

Translation Parsing is an important step in translations based on the transfer between language at a deep abstract syntactic level (or possibly at a semantic level).

Among these various application domains, ATOLL focuses its efforts on knowledge acquisition and text mining, in particular through the action BIOTIM for processing botanical corpora (cf. 7.3) and new action PASSAGE (cf. 7.1).

5. Software

5.1. System Syntax

Participants: Pierre Boullier [maintainer], Philippe Deschamp, Benoît Sagot.

SYNTAX *on INRIA GForge: https://gforge.inria.fr/projects/syntax/*

The (not yet released) version 6.0 of the SYNTAX system has been extended and now includes SXSPELL, a spelling error corrector and SXLFG a Lexical Functional Grammar processor which is divided in two main parts (Section 6.1) : the constructor part which compiles the LFG specifications and the parser part which processes a source text w.r.t. these compiled specifications.

This version of SYNTAX runs on various 32bit platforms such as Linux, Solaris, HP/UX and Windows. A first 64-bit port has been made for HP/UX. Optimized ports for 32-bit compatible 64-bit architectures are currently in progress, including 64-bit x86 running Linux and IBM G5 running Mac OS X.

Release 3.9 essentially handled deterministic CFGs of type LALR(1). Release 6.0 extends it by including RLR (an extension of LR parsing strategy in which an unbounded number of look-ahead terminal symbols may be used, if necessary), non-deterministic CF parsers based upon push-down automata of type LR, RLR or left-corner, and a parser generator for Range Concatenation Grammars (RCGs), hence the leap in numbers from 3 to 6.

SYNTAX has recently be transferred on INRIA GForge.

5.2. System DyALog

Participant: Éric Villemonte de la Clergerie [maintainer].

DYALOG *on INRIA GForge: http://dyalog.gforge.inria.fr/*
DYALOG*: http://atoll.inria.fr Rubrique « Logiciels »*

DYALOG provides an environment to compile and execute grammars and logic programs. It is essentially based on the notion of tabulation, i.e. of sharing computations by tabulating traces of them. DYALOG is mainly used to build parsers for Natural Language Processing (NLP). It may nevertheless be used as a replacement for traditional PROLOG systems in the context of highly ambiguous applications where sub-computations can be shared.

The current release **1.11.2** of DYALOG is freely available by FTP under an open source license and runs on Linux platforms for x86 architectures and on Mac OS intel. A port for PowerPC, initiated by Djamé Seddah, should be soon available.

The current release handles logic programs, DCGs (*Definite Clause Grammars*), FTAGs (*Feature Tree Adjoining Grammars*), FTIGs (*Feature Tree Insertion Grammars*) and XRCGs (*Range Concatenation Grammars* with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [15].

C libraries can be used from within DYALOG to import APIs (`mysql`, `libxml`, `sqlite`, ...).

DYALOG is largely used within ATOLL to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASY (cf. 7.4 and [13]).

DYALOG is also an essential component in the development of a robust Portuguese parser at the New University of Lisbon. It is occasionally used at LORIA (Nancy), University of Coruña (Spain) and University of Pennsylvania.

DYALOG and other companion modules have recently be transferred on INRIA GForge.

5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de la Clergerie [correspondant].

MetaGrammar Toolkit on INRIA GForge: http://mgkit.gforge.inria.fr/
MGCOMP, MGTOOLS, *and* FRMG*: http://atoll.inria.fr Rubrique « Catalogue »*

DYALOG (cf. 5.2) has been used to implement MGCOMP, a compiler of Meta-Grammar (cf. 6.6). Starting from an XML representation of a MG, MGCOMP produces an XML representation of its TAG expansion.

The current version **1.4.1** is freely available by FTP under an open source license. It is used within ATOLL and (occasionally) at LORIA (Nancy) and at University of Pennsylvania.

The current version adds the notion of namespace, to get more compact and less error-prone meta-grammars. It also provides other extensions of the standard notion of Meta-Grammar in order to generate very compact TAG grammars. These extensions include the notion of *Guarded nodes*, i.e. nodes whose existence and non-existence depend on the truth value of a guard, and the use of the regular operators provided by DYALOG on nodes, namely disjunction, interleaving and Kleene star.

The current version of MGCOMP has been used to compile a wide coverage Meta-Grammar FRMG to get a grammar of around 100 TAG trees [13]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available. The current version (1.0.2) of FRMG has been completed to handle *support verbs* (such as *prendre garde [à]*) and to take into account some recent modifications of LEFFF 2.

To ease the design of meta-grammars, a set of tools have been implemented by É. de la Clergerie and F. Thomasset, and collected in MGTOOLS (version **2.0.0**). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views. A new version is under development to provide an even more compact syntax and some checking mechanisms to avoid frequent typo errors.

The various tools on Metagrammars have recently been transferred on INRIA GForge.

5.4. Linguistic Workbench

Participants: Benoît Sagot, Éric Villemonte de la Clergerie, Pierre Boullier, Isabelle Cabrera.

ATOLL Linguistic Workbench on GForge: http://lingwb.gforge.inria.fr/
List of tools: http://atoll.inria.fr Rubrique « Catalogue »

ATOLL develops several tools that may be used for the first levels of linguistic processing preceding parsing, in particular morpho-syntax. They are freely available under open source licenses, keeping in mind that most of these tools are still beta versions.

SXPIPE (1.1.0) a container package, developed by B. Sagot, that includes many scripts for morpho-syntactic processing. It also includes a spelling corrector (SXSPELL) and a segmenter (for sentences and tokens) that rely on SYNTAX. The deployment of the various component is handled by LINGPIPE.

LINGPIPE (0.2.0) a small set of Perl modules originally developed by É. de la Clergerie to setup and configure a linguistic pipeline. The current version of lingpipe comes with a basic set of wrappers for the various linguistic tools we use for the morpho-syntactic processing of French (tokenizer, tagger, lexicon lookup, ...)

LEXED (4.6) a C software originally developed by L. Clément to build efficient and compact lexica from lists of words (completed with additional information).

Other tools are also developed to deploy our linguistic processing chain, run it on clusters, get statistics, ...

Most of these tools have been recently transferred on INRIA GForge. The others should follow on completion of their packaging.

5.5. Lexicon Lefff

Participant: Benoît Sagot.

French morphological lexicon LEFFF*: http://www.lefff.net*
Alexina on INRIA GForge: http://gforge.inria.fr/projects/alexina/

LEFFF 1 is a freely available French morphological lexicon for verbs that has been automatically extracted from a very large corpus.

A new version of LEFFF, LEFFF 2 version 2.2.1, used in ATOLL and freely available, covers all grammatical categories (not just verbs) and includes syntactic information (such as verb categorization frames).

LEFFF and various tools to acquire and manage Lefff-like lexica have recently been transferred on INRIA GForge.

6. New Results

6.1. Contextual Parsing with LFGs

Keywords: *Context-sensitive grammatical formalisms*, *grammatical modularity*, *lexical functional grammars*, *polynomial parse time*, *shared parse forests*.

Participants: Pierre Boullier, Benoît Sagot, Alexis Nasr.

LFG *Lexical Functional Grammars*

This year, our work mainly concentrates on the improvement of our Lexical Functional Grammar [LFG] parser SXLFG. In particular, we developed a more sophisticated architecture which enables the computation of different kind of modules, including a (rule-based) chunker filtering module and an endogenous probabilistic n-best filtering module. We also extended the panel of available operators, hence enriching the syntax that may be used to write grammars for SXLFG, without slowing down the parsing process.

Lexical Functional Grammar is a grammatical theory assuming two parallel levels of syntactic representation: constituent structure (c-structure) and functional structure (f-structure).

- C-structures have the form of context-free phrase structure trees;
- F-structures are sets of pairs of attributes and values; attributes may be features, such as tense and gender, or functions, such as subject and object.

At least at a conceptual level, we may see an LFG parser as a two-phase process: the first phase is a CF parser which builds the C-structure while the second phase evaluates the F-structure on the tree built by the first phase. However, the CF-backbone of real linguistic grammars (including LFG) are usually massively ambiguous. For example, for a sentence, we have exceeded the capacity of a single floating point 32 bit word in counting its number of parse trees. In ATOLL, we know how to handle such a combinatorial explosion of resulting tree structures. In the LFG context, this means that, for any given sentence w, we can compute in polynomial time a polynomial size parse forest which represents all the possible C-structures of w (See for example [5]). However, the efficient evaluation of F-structures on parse forests is still a research problem. Of course, the unfolding of the parse forest into single trees upon which F-structures are evaluated is not a viable method. We have designed and implemented a method which evaluates F-structures directly on a parse forest and which shares common [sub-]computations.

The coupling of our guided Earley parser with the previous shared computation of F-structures results in a new LFG parser called SXLFG. It is able to handle cyclic F-structures, implements a lazy unification to optimise the computation of these structures, and allows the grammar writer to specify disambiguation heuristics that can be applied on F-structures associated with any node of the forest. This improvements w.r.t. the first version of SXLFG have resulted in parsers that run approximately 5 to 10 times faster.

But this year's improvements have extended the scope of SXLFG. Indeed, a rule-based chunker module has been developed. When used alone in a deterministic way on top of the c-structures forest, it leads to a very efficient chunker, whose precision results are state-of-the-art. We also developed an n-best filtering module that relies on probabilistic data which are themselves learned from a previous execution of SXLFG on a corpus (cf.6.2). These different modules, as well as an extended output module, can be used in a procedural way, hence allowing, for example, to use efficient filtering methods before trying to compute f-structures, and then, in case of failure, to go back to earlier states of the c-structures forest, to retry f-structure computation on a less-filtered forest.

Though this parser still needs to be improved, it is sufficiently mature to support full natural language descriptions. SXLFG is one of the three parsers used by ATOLL in the EASY campaign (cf. 7.4). It has been used as well to parse large (multi-million word) corpora, in order to validate our parsing techniques but also to learn information from the resulting analyses.

6.2. N-best parsing

Keywords: *disambiguation*, *n-best parsing*.

Participant: Pierre Boullier.

The keys for efficient parsers may mainly be partitioned in two complementary paradigms: Computation sharing and reduction of the search spaces. In the ATOLL project, these two approaches have been largely investigated by the past and are largely responsible for the high quality of our tools. Dynamic programming methods illustrate the first approach, while guiding techniques may be seen as a way to shrink the search space of items during chart parsing. The conjunction of these two techniques allows to build full shared parse forests of NL sentences in a very reasonable time, though the number of individual trees (parses) in such forests may be gigantic (we have reach numbers that are close to the estimated number of particles in the universe!). Of course, post processors for chart parsers, such the SXLFG F-structures constructor, must be able to deal with such huge forests. Once again, the computation sharing techniques give good results for forests with up to say 10^{10} or 10^{20} individual trees. But larger trees still stay out of reach in reasonable times (say a very small number of seconds).

This year, we have tried to investigate the pruning of parse forest search space by statistical methods. Unfortunately, we may admit that all the parse trees in a forest have not the same level of validity. Some must be clearly rejected (they occur in a forest because the syntactic backbone is unavoidably too permissive in order to keep the grammatical description within a manageable size), while the others are valid, though some of them may be considered as being near the borderline of the processed language. In other words, we assume that it exists (real) numbers called *weights* that can be associated with each tree and which reflect, using the $<$ relation, the fact that a tree is more credible than another one. Moreover, we assume that the weight of a tree may be computed from the weights of its subtrees. The choices for the *best* trees are twofold. On the one hand, for a given integer value n, we may want to choose the n best trees or, in the other hand, for a given real threshold value v, we may want to choose all trees whose weight is greater or equal to v. We have designed and implemented a single algorithm which handle these two facets as well. Of course, for combinatorial explosion reasons, such an algorithm can neither unfold the forest into single trees nor compute the weight of each tree.

If we only consider the n-case, the first idea that came at mind is to compute for each sub-tree the list of its n-best weights and to propagate these values bottom-up towards the root of the forest. This is not very nice since the number of sub-computations must be much greater than necessary. Moreover this vision is hardly compatible with the threshold v-case. In order to deal with both the n and v-case, we have designed an incremental algorithm. This algorithm computes in a first path over the forest the best tree, in a second path the second best tree, ..., and in a p-th path the p-th best tree. In the n-case we can stop when $n = p$, while in the v-case, we stop when the weight of the p-th tree is less than v.

In fact, each path computes the weight of its dedicated tree in using information (excepted for the first path) that have been computed by the previous path and in storing information's that can be used by the next path, together with in-formations that may allow to prune the forest. Note that this algorithm does not compute any unnecessary sub-weight. Moreover, it proves to be efficient: we may compute and prune large forests for values of n of several hundreds of thousand in few seconds. We have choose to keep the global structure of the forest in only cutting dead-ends branches (that are not shared with any n-best trees). This option has several advantages but, in general, the number of trees in this pruned forest is greater that n.

However, this computation can only take place if each atomic sub-tree possesses an initial weight. Recall that these atomic sub-tree are instantiations of CF productions. In other words, the whole process can only work if we have a probabilistic or weighted CFGs. In the context of SXLFG, the weights of this CFG have been automatically computed as follow. A corpus of several hundred thousand of sentences have been parsed with our SXLFG parser which, for each sentence, has output its most credible (in the sense of the underlying LFG grammar) parse. It is thus possible to count the number of times a given production is used in a parse and to sum these values over the corpus. After normalization (the weight of an A-production is the number of occurrences of that production over the total number of occurrences of all the A-productions), we get a weighted CFG that can be used to compute the weight of any parse tree. Note that we have extended this model of weighted CFG in associating with each production not a single value but a vector of weights. Each weight in a vector expresses the validity of the associated production in a given context. The single value expresses no context. For example a context for a production p may be something like "p occurs as the third son of the non-terminal A and is such that its first son is empty". Of course, for each available context, the n-best algorithm must be able to handle it. This can be non-trivial, particularly when the information is not strictly bottom-up.

As suggested at the beginning of this section, a possible application of our n-best algorithm is to prune the parse forest in an intermediate phase between a CF parser and an F-structures evaluator. However, to do that, for a given sentence, we must know which value n to apply. In order to have an idea of n, we perform the following measures. Assume that, as previously explained, we have computed our weighted CFG. It is thus possible, for any given sentence processed by SXLFG to compute the weight v of its single F-structure decorated parse tree. If the parse forest of this sentence reparsed by the weighted CF parser is at turn pruned by the threshold v-best algorithm, it is possible to know the rank n of this parse tree. On a given corpus, it is thus possible to compute a function $\mathcal{N}$ which, for example, gives the value $n = \mathcal{N}(l, p)$ for a sentence of length l for a probability p (i.e., the probability, for our sentence of length l, that its unique parse tree — in the sense of SXLFG — is among the n-best trees is p).

However, preliminary results that must be confirmed, suggest that, on long sentences, if a high probability p is seeked, the computed value n is so high that the whole pruning process takes as much times as the F-structures computation evaluated on the unpruned parse forest.

We can note that the previous researches may also be seen as a way to evaluate how a LFG can be approximated by its underlying CFG augmented with a weight mechanism.

In a similar vein, we have implemented, but not yet evaluated, a statistical ranking mechanism of F-structures. If we consider the single tree decorated with its F-structures produced by our SXLFG parser on a sentence, in addition to the number of occurrences of productions, we may output, for each production occurrence a signature for its associated F-structure. This signature is a finite structure which summarizes the (recursive) content of this F-structure. From an input corpus, it is thus possible to compute, for each (selected) production a list of the most probable F-structure signatures. This list can thus be used, at the LFG designer will, to prune the list of already computed F-structures in such a way that the processing can go on with a lower number of F-structures.

6.3. Stochastic Parsing

Keywords: *grammar extraction*, *stochastic parsing*.

Participants: Alexis Nasr, Pierre Boullier.

INRIA-TALANA parser: http://www.lif-sud.univ-mrs.fr/~nasr/gdg

TIG *Tree Insertion Grammars*

This work pursued a joint work initiated by Alexis Nasr at University Paris 7 and Owen Rambow at Columbia University [45], [46], [44] [20], [21]. The objective is to produce a stochastic parer from an automatically extracted from a treebank. A first version of the parser has been obtained and evaluated on reference corpora (Penn Treebank and Paris 7 Treebank). However, this prototype suffered from efficiency problems, both in time and space.

During his "delegation" at INRIA in ATOLL, Alexis Nasr has reimplemented the parser using SYNTAX. The development of this new parser has required a transformation of the underlying grammars, from the original *Tree Insertion Grammars* [TIG] to Context-Free Grammars (TIGs being equivalent to CFGs). Secondly, the search algorithm for the most probable parse tree in the output shared forest produced by the original parser had to be adapted to the kind of forests produced by SYNTAX. The new parser is significantly more efficient than the original one. More precisely, the efficiency gain has allowed us to increase our search space for the most probable parse, which resulted into an increase of 5% in precision, from 80% to 85% (the precision is defined as the ratio of correct dependencies in the most probable parse). The new INRIA-TALANA parser is available on line. It has been evaluated at Columbia University and at AT&T Labs. The results will be published in [21].

6.4. Spelling error correction

Keywords: *finite state transducer*, *spelling correction*.

Participants: Benoît Sagot, Pierre Boullier.

Following the development of our spelling correction techniques from 2004, based on finite transduction techniques, we have improved in 2006 the quality of our spelling rules in SXSPELL.

But more importantly, we have extended SXSPELL's architecture, in order to enable the support of different languages. Indeed, we now have a Polish version and a preliminary Slovak version of SXSPELL, which are used in the corresponding versions of SXPIPE.

6.5. Automata and Tabulation for Parsing

Keywords: *Dynamic Programming*, *Parsing*, *Tabulation*, *Thread Automata*.

Participant: Éric Villemonte de la Clergerie.

TAG *Tree Adjoining Grammars*

TA *Thread Automata*

MCS *Mildly Context-Sensitive formalisms*

TWA *Tree Walking Automata*

TWT *Tree Walking Transducer*

At a theoretical level, we are still trying to understand the full power of Thread Automata [TA], and, possibly, to extend them. TA have originally been introduced to ensure dynamic programming interpretations for Mildly Context-Sensitive [MCS] formalisms. We have explored their relationships with the better known class of (deterministic) Tree Walking Automata [TWA], transducers [TWT], and extensions with pebbles. Our preliminary conclusion is that there seem to be equivalent to deterministic Tree Walking Transducers, which is coherent with previous results by David Weir about the equivalence of TWT and MCS [49].

Unfortunately, TAs are not powerful enough to cover some extreme cases of scrambling (that are outside the scope of MCSs), as illustrated, for instance, with the MIX language of all sequences on alphabet a, b, c with an equal number of occurrences for each letter ($\{w \in \{a,b,c\}^* \| |w|_a = |w|_b = |w|_c\}$). We tried to design an extension of TAs, namely *Shared TAs*, allowing a thread to be shared by several parents and powerful enough to capture these kinds of scrambling phenomena and also syntactic sharing phenomena (such as control verbs). Shared TAs look similar to Graph Walking Transducer with conditions on the topology of the underlying graph. However, several details remains to be checked.

We also thinking about the implementation of TAs within DYALOG and at their use, in conjunction with Meta-Grammars, for handling Multi-Component TAGs.

6.6. Designing grammars using Meta-Grammars

Participants: Éric Villemonte de la Clergerie, Julien Martin.

MG *Meta-Grammars*

The exact formalization of Meta-Grammars (MG) is still a subject of research that we explore through cooperations with Project-Teams "Langues & Dialogue" and "Calligramme" (LORIA), in particular through INRIA ARC MOSAIQUE (cf 7.2).

Roughly speaking, a meta-grammar is a list of classes expressing constraints. A class may inherit constraints from one or more parents and is used to describe some elementary linguistic phenomena. Constraints express existence of nodes, relationships between these nodes (ancestor, parent, sibling, equality, ...) and content as feature structures attached to nodes or to the class. A class can also states that it provides or needs some functionality. The role of a MG compiler is to combine classes in order to get neutral classes (all needs filled by providers and conversely), to check that constraints are satisfied and to use these constraints to generate the (minimal) structures of the grammars (trees in the case of TAGs).

É. de la Clergerie has developed, with DYALOG, a prototype of MG compiler, called MGCOMP (cf. 5.3). This prototype allows the exploration of new features for Meta-Grammars. During his master internship [34], Julien Martin has explored several possible couplings of MGCOMP (and therefore DYALOG) with external constraint solvers, in order to explore the benefice of powerful domains of constraints for Metagrammars. The particularities of MGs, namely the large number of class combinations, have shown the difficulties of this approach, because constraint solvers are generally designed to solve one large constrained problem rather that a large number of related small problems. Still, the coupling of DYALOG with SMODELS (based on the resolution of Datalog stable models) seems potentially very interesting, for MGs but also for other tasks.

6.7. Acquisition of morphological and syntactic lexical information

Participant: Benoît Sagot.

French morphological lexicon LEFFF*: http://www.lefff.net*

Among the different resources that are needed for Natural Language Processing tasks, the lexicon plays a central role. However, the development or enrichment of a large and precise lexicon, even restricted to morphological information, is a difficult task, in particular because of the huge amount of data that has to be collected. Therefore, most large-coverage morphological lexicons for NLP concern only a few languages, such as English. Moreover, these lexicons are usually the result of the careful work of human lexicographers who develop them manually over years, and for this reason they are often not freely available.

Therefore, we currently investigate methods to automatically acquire lexical knowledge, in particular morphological and syntactic knowledge [25]. These methods, that may involve manual validation to guarantee the quality of the resources that are produced, had been successfully applied to supplement the LEFFF lexicon for French in 2004, and have been used to acquire from scratch a lexicon for Slovak [12], language that lack large-coverage resources. Our method involves now also derivational morphology, which is a link to the acquisition of syntactic knowledge. In 2006, it has been used to extend an already existing morphological lexicon for Polish, so as to improve the morphologically-annotated corpus of the Institute of Computer Science of the Polish Academy of Science.

Direct acquisition of syntactic knowledge has been also performed for French and used to add syntactic information to the new version of the LEFFF, concerning in particular verbal lemmas. This new version, the LEFFF 2 [11] [28], is now a large-coverage formalism-independent syntactic lexicon for French, and is currently used in all our parsers.

But the LEFFF has also been extended thanks to the exploitation of linguistic information which is available in other resources, in particular for impersonal verbs and frozen verb phrases.

Recent advances in the lexical formalism itself used by the LEFFF have led to a distinction between syntactic functions and their realizations. This has enabled preliminary work to compare the LEFFF to the two other available resources for French, namely the lexicon-grammar tables (under their SynLex form) and the Proton (now Dicovalence) lexicon.

6.8. NLP Infrastructure and standardization

Participants: Benoît Sagot, Isabelle Cabrera, Éric Villemonte de la Clergerie.

French morpho-syntax demo: http://atoll.inria.fr/mafdemo

ATOLL tries to design and setup an full processing chain for French, with various components relying on XML representations. The resulting pipeline SXPIPE [10] covers the first layers of linguistic processing, namely morpho-syntactic processing (segmentation, tagging, lexicon lookup, named entities, ...), cf. 5.4.

As for SXSPELL, SXPIPE's architecture has been extended in order to enable the support of different languages. We now have a Polish version and a preliminary Slovak version of SXPIPE. The Polish version, developed during a 3-month stay of B. Sagot in the Institute of Computer Science of the Polish Academy of Science (Warsaw), has been used to improve the quality of the morphosyntactically annotated corpus developed there.

The main role of SXPIPE is to provide input to the parsers developed by ATOLL. At the other end of the chain, we are developing various tools to run our chain on large corpora using clusters, to get various statistics, to get syntactic output as shared dependency forests, and to disambiguate these forests.

With the arrival in ATOLL of Isabelle Cabrera as an engineer, the set of involved tools and linguistic resources being developed by ATOLL has been migrated to INRIA GForge. A script has been developed to ease the automatic deployment, compilation, and installation of the full ATOLL processing chain, in particular on the nodes of a cluster. Another issue concerns the possibility to easily configure the processing chain for a specific corpus (specific raw format and metadata, specific lexica, named entities, acronyms, grammar, ...).

6.9. Evaluation

Keywords: *Evaluation*, *Parsing*.

Participants: Éric Villemonte de la Clergerie, Benoît Sagot, Alexis Nasr.

EASY *Action: http://atoll.inria.fr Rubrique « Projets »*
Demo on error mining: http://atoll.inria.fr/perl/results5/errorscgi.pl

ATOLL has developed several important linguistic resources and tools, such as LEFFF, SXPIPE, FRMG, and SXLFG. More and more, we need to evaluate and assess their quality before going further.

The participation of ATOLL to the French evaluation campaign EASY (in December 2004) has been a first step in this direction. The participants to EASY were expected to return information about 6 kinds of non recursive constituents (Nominal chunks, Adjectival chunks, Adverbial chunks, verbal kernels, ...) and 14 kinds of dependencies (verb-subject, verb-object, ...). The evaluation was done on a set of around 35000 sentences covering various kinds of style (journalistic, literacy, mail, medical, speech, questions).

We are still waiting for the full results of this campaign, but have already received preliminary results about constituents for a subset of 4262 sentences and preliminary results about dependencies. We have started using these results to analyze the weaknesses of ours tools and resources and have developed a few scripts to be able to synthesize our own statistics. We are now able to replay the EASY experiment and have actually started to do it. Our new results already show a clear improvement w.r.t. the original campaign (statistics for FRMG at http://atoll.inria.fr/results6/index.html).

We are also testing our parsers (FRMG and SXLFG) on other corpora, in particular a journalistic corpus "*le monde diplomatique*" (17Mwords) and some of the botanical corpus used for BIOTIM. More than 300 Ksentences have been parsed with FRMG, with a coverage rate (for full parsing) of 42% and more statistics may be found at http://atoll.inria.fr/results5/distrib.html. Almost 400 Ksentences have been parsed with the new version of SXLFG, with a coverage rate of 53%, although the precision of these parses are slightly lower than those of FRMG [25].

We have also developed error mining techniques to find, in the parsed corpora, the words that have significantly low parsability rates. Such words usually denotes incorrect or incomplete entries in our lexicon LEFFF, or errors in the grammar. The idea, presented in [31], [30], is based on an iterative converging algorithm that collects local information (at sentence level) to compute global statistics (at corpus level) and re-inject these statistics locally. This algorithm is completed by an WEB interface that allow an human to browse words, ranked by their error rate, and emit comments. The algorithm was tested on our results on "*le monde diplomatique*" for both FRMG and SXLFG. It was also useful in the context of the action BIOTIM (cf 6.10).

Mining errors in large corpora allows to us to identify for each suspect a set of sentences where the suspect is the prime one explaining their non parsability. In his master internship [35], Lionel Nicolas has explored ways to propose corrections for the suspects. For a suspect word w, the idea is to generalize the lexical information attached to w (to get a kind of unknown word) and re-parse the sentences associated to w. The experiments are still preliminary. However, the first results show an high increase of the parsability rate with these generalized words. We also showed that a restricted set of syntactic constructions for w may identified over the set of sentences. It seems therefore possible to propose for human validation new lexical information for the suspect w. However, it remains sometimes difficult to differentiate syntactic constructions that have a tendency to occur in similar contexts (for instance, it is difficult to differentiate an adjective from a past-participle or sometimes, a noun from an adjective).

6.10. Processing Botanical Corpora

Participants: Éric Villemonte de la Clergerie, Milagro Fernandez Gavilanes, Olg'a Feiguina.

BIOTIM *Action: http://atoll.inria.fr Rubrique « Projets »*

In the context of French action BIOTIM (cf. 7.3), ATOLL is involved in processing botanical corpora.

After the initial steps dealing with terminological extraction and logical structuring of the corpora, we tuned a specific version of our processing chain to parse these corpora and extract some knowledge.

By running 14 rounds, we progressively improved the parsing coverage from 36% to 68% on around 80000 sentences. These results were obtained by:

- Using our error mining techniques (cf. 6.9) to quickly get feedback about most frequent errors;
- By tuning a specialised metagrammar, derived from the general FRMG, where some syntactic phenomena have been deactivated while missing ones have been added.
- By improving the pre-parsing phases (segmentation, named entities detection, ...).

The resulting syntactic output represented by highly ambiguous shared dependency forests (encoded as XML files) have been exploited by Milagro Gavilanes to extract semantic classes (for instance the classes of plant parts, colors, textures, shapes, ...). A first step was to adapt the ideas behind error mining to perform partial disambiguation. By iterating between sentence level and corpus level, it is possible to learn the most probable syntactic contexts for a given word. The second step is to group words into semantic classes, based on the

assumption that words occurring into similar syntactic contexts belongs to a same class (*Harris distributional hypothesis*). This grouping into classes is also helped by using seed words and by identifying good linguistic markers (such as "*en forme de*" to detect shapes) and constructions (such as range constructions "*X à X*"). Part of this work is going to be published in [27]. The kind of information being extracted will help us building a small "ontology" that will identify for instance that a specific color or shape is pertinent for leaves but not for fruits, extending preliminary results obtained through a collaboration with François Role (Univ. Paris 5) [22].

At sentence level, Ol'ga Feiguina has developed a Perl prototype of n-best disambiguator taking a shared dependency forest as input and returning a partially disambiguated shared dependency forest.

6.11. Morphology and finite state transducers

Participant: François Barthélemy.

François Barthélemy worked on Finite-State Morphology, the approach of Natural Language Processing which consist in describing the morphology of human languages into various formalisms which are compiled into finite-state machines (automata and transducers). There are two main approaches:

- describing arbitrary relations between word components and actual forms using cascading contextual rewrite rules;
- describing same-length relations, where each symbol of the abstract component is mapped to exactly one symbol of actual forms and conversely, using two-level rules applied in parallel.

Following previous propositions (e.g. the work by Kiraz [43]), we investigated an intermediate approach where there is a mapping between substrings instead of single symbols, within the tuples of the relations. We defined a new class of transducers which is closed under intersection, whereas arbitrary transducers are not . Intersection allows for modular descriptions and the morphology of a language may therefore be described as the intersection of local constraints. The transducers that are used are n-tape transducers which describe n-ary relations. They have the same theoretical power as Finite-State Automata, but are more convenient to describe morphology. We are currently working on some applications which demonstrate the interest of the formalism [24].

6.12. Implicit Information in Natural Language

Participant: Areski Nait Abdallah.

Areski Nait Abdallah continued his work on the potential use of partial information logics to handle natural language pragmatic, work initiated through joint publications with Alain Lecomte. In particular, he has started an OCAML implementation of his partial information logic.

He has also worked on the use of a typed lambda-calculus for an algebraic approach of his logic.

6.13. Free Software

Keywords: *Copyright*, *Economy*, *Free Software*, *Linux*, *Open Source*, *Patent*.

Participant: Bernard Lang.

The problem raised by the open availability of linguistic resources, whether linguistic processing software (such as taggers, parsers, etc.) or linguistic data (such as lexicons, grammars, or corpora) has raised our interest in the development of free scientific resources. There is a wide consensus that the limited availability of the results produced by earlier research, due to excessive use of intellectual property, has been a major impediment to the progress of computational linguistics research, especially in Europe.

It is a policy of our group to make our results freely available.

B. Lang has taken a strong interest in these issues and has become very active in understanding better the legal and economic aspects of the production, dissemination and use of intangible goods. Much of the work is observing the evolution of the free economy of intangibles, how it develops, and how it relates to the evolution of the legal system. One important aspect is the impact on research practice, on communication between researchers, and on the valorization of research results.

7. Contracts and Grants with Industry

7.1. Action PASSAGE (2006 – 2008)

Participants: Éric Villemonte de la Clergerie, Pierre Boullier.

PASSAGE *EASy homepage: http://www.limsi.fr/Recherche/CORVAL/easy/*

PASSAGE is an action recently accepted by the ANR MDCA program (*Masse de Données Connaissance Ambiantes*) and should start beginning of 2007. The participants are ATOLL (coordinator), LIR (LIMSI, Orsay), "Langue & Dialogue" (LORIA, Nancy), LI2CM (CEA-LIST), plus several contractors (ELDA, TAGMATICA and several providers of parsing systems).

PASSAGE stands for "*Large Scale Production of Syntactic Annotations to move forward*" . Its main objectives are to parse a large corpus (100 to 200 million words) with several parsers (around 10 systems), combine the results provided by these parsers and use the resulting annotations to acquire new linguistic knowledge (semantic classes, subcategorization frames, disambiguation probabilities, ...). A small part of the corpus (around 400000 words) will be manually validated to be used as a reference treebank. Two evaluation campaigns based on the work done during the Technolangue action EASY (cf. 7.4) will be conducted during PASSAGE to assess the performances of the parsing systems. The annotations and derived linguistic resources will be made available.

7.2. Action MOSAIQUE (2006 – 2007)

Participants: Éric Villemonte de la Clergerie, Benoît Sagot.

MOSAIQUE *Homepage: http://mosaique.labri.fr/*

MOSAIQUE is an INRIA ARC (*Action de Recherche Coopérative*) which groups the INRIA teams working in Computational Linguistics (ATOLL, "Langue & Dialogue"/LORIA, Calligramme/LORIA, Signes/Futurs Bordeaux), Lattice (TALaNa, Paris 7), MODYCO (Paris 10), LLF (Paris 7), and LPL (Univ. of Provence). It is coordinated by Lionel Clement from Signes.

The participants of this ARC are involved in the development of linguistically motivated grammars relying on various formalisms and theories. Inspired by our experiences on Metagrammars, MOSAIQUE's objectives are to design an high level syntactic model that may then be compiled (possibly with approximations) into our various "operational" target formalisms (TAG, LFG, HSPG, Interaction Grammars, ...).

7.3. Action BIOTIM (2003 – 2006)

Participants: Éric Villemonte de la Clergerie, Benoît Sagot.

BIOTIM *home page: http://www-rocq.inria.fr/imedia/biotim/*

Funded by ACI program on "Masses de données" (Data Warehouses), action BIOTIM has started end of 2003 for 3 years. Its thematic is the processing of botanical textual corpora and image collections in order to extract knowledge and establish bridges between texts and images for more intelligent navigations at a semantic level. ATOLL is essentially concerned with the linguistic processing of textual corpora with generic methods to extract terminologies, ontologies and knowledge bases.

The other participants to BIOTIM are INRIA project-team IMEDIA (leader), CNAM team Vertigo, INRA team URGV, IRD, and LIFO (University of Orléans).

7.4. Action EVALDA/EASY

Participants: Éric Villemonte de la Clergerie, Pierre Boullier, Benoît Sagot.

EASy Home page http://www.limsi.fr/Recherche/CORVAL/easy/

ATOLL has participated to the parsing evaluation campaign EASY of action EVALDA of French program Technolangue. The campaign took place on mi-december 2004 with the participation of 14 parsers. The participants to EASy were expected to return information about 6 kinds of non recursive constituents (Nomimal chunks, Adjectival chunks, Adverbial chunks, verbal kernels, ...) and 14 kinds of dependencies (verb-subject, verb-object, ...). The evaluation was done on a set of around 35000 sentences covering various kinds of style (journalistic, literacy, mail, medical, speech, questions).

ATOLL has provided results for two parsers, namely FRMG and SXLFG [7]. We have received a preliminary evaluation for the constituents on a subset of 4262 sentences. We are still waiting for a complete set of results from the organizers, in particular regarding the dependencies.

7.5. Action LexSynt (2005 – ??)

Participants: Benoît Sagot, Éric Villemonte de la Clergerie.

LEXSYNT *Home page http://lexsynt.inria.fr/*

The LEXSYNT action, funded by ILF (*Institut de Linguistique Française*), groups 13 teams, including INRIA teams ATOLL, Calligramme, "Langue & Dialogue", and Signes. The main objective of this action is to design a reference syntactic-semantic lexicon for French. The work should take place in coordination with existing producers of lexicons (for merging resources), with grammar designers (to ensure usability in parsers), and with the current proposal **LMF** for the standardization of lexicons (ISO TC37 SC4).

8. Other Grants and Activities

8.1. National Actions

Ph. Deschamp is a member of the French "Commission spécialisée de terminologie de l'informatique et des composants électroniques" (terminology committee for Computer Science and Electronic), and distributes on-line the glossary http://www-rocq.inria.fr/who/Philippe.Deschamp/CMTI/ resulting of his work (more than 130 000 downloads). Ph. Deschamp is also a member of the French "Commission spécialisée de terminologie et de néologie des télécommunications" (terminology committee for telecommunication).

B. Lang is vice-president of AFUL (http://www.aful.org), "Association Francophone des Utilisateurs de Linux et des Logiciels Libres", and member of the administration board of ISoc-France, the Internet Society French branch. He is also a member of the scientific board of association SOISSON Informatique Libre.

8.1.1. Open Source Software

B. Lang has presented the notion of open source software in several workshops, talks and conferences, organized by local collectivities and administrations.

8.2. International networks and working groups

8.2.1. Open Source Software

B. Lang has been several times invited to talk on Open Source Software.

B. Lang is a member of an expert committee on Open Source Software for the European Commission General Direction for Information Society (ex DG 13) (http://eu.conecta.it/).

8.2.2. PAI Pessoa KLING (2005 - 2006)

Funding for visits has been granted by the French-Portuguese PAI (Programme d'actions intégrées) PESSOA to continue a long-lasting collaboration between ATOLL and team CENTRIA of Lisbon New University, led by Gabriel Pereira Lopes. In 2006, this program was used for the visit of Vitor Rocio.

8.2.3. Former PAI PICASSO CATALINA-2

For administrative reasons, it is no longer possible to submit a new French-Spanish PAI (Programme d'actions intégrées) PICASSO between ATOLL and team COLE at Universities of La Coruña and of Vigo, led by Manuel Vilares Ferro. However, we have continued a program of visits, with a long visit by Francisco Jose Riberra Pena, from November 2005 to March 2006, the internship of Milagro Gavilanes (cf. 6.10), and an invitation of É. de la Clergerie at Universities of Santiago, Coruna, and Vigo (March).

8.2.4. XTAG Collaboration

We have renewed some contacts with the group XTAG at University of Pennsylvania, in relation with Meta-Grammars.

8.2.5. ISO subcommittee TC37 SC4 on "Language Resources Management"

The participation of ATOLL to French Technolangue action Normalangue has resulted in a strong implication in ISO subcommittee TC37 SC4 on "Language Resources Management" (http://www.tc37sc4.org/). É. de la Clergerie has participated to ISO events (Berlin, April 2005; Warsaw, August 2005) and has played a role of expert (in particular on morpho-syntactic annotations [MAF], feature structures [FSR & new FSD], and on the new work item on syntactic annotations [SynAF]).

8.3. Visits and invitations

A 3 months visit of Francisco Jose Ribadas Pena (Univ. of La Coruna) from November 2005 to March 2006.

A one week visit of Vitor Rocio (Univ. of Lisboa), December 2006.

9. Dissemination

9.1. Animation at INRIA

B. Lang was an elected member of INRIA's "Conseil Scientifique" (till November), an elected menber of INRIA's board ("conseil d'administration"), a member of CUR (Research Unit Committee), and a member of CSI ("Comité de Suivi des Ingénieurs").

É. de la Clergerie is member of the GTAI subcommittee of COST committee and an elected substitute member of INRIA's "Conseil scientifique".

9.2. Supervising

É. de la Clergerie co-supervised the PhD thesis of Benoît Sagot with Laurence Danlos (TALaNa/LATTICE, University Paris 7). He has also supervised the internships of :

- Julien Martin (Univ. Paris 6) on coupling Metagrammars and constraint solving (cf. 6.6) [34]
- Ol'ga Feiguina (Univ. of Montreal) on the disambiguation of shared dependency forests (cf. 6.10),
- Lionel Nicolas (Univ. of Nice, co-supervising with Jacques Farré) on proposing lexical corrections (cf. 6.9) [35].

9.3. Jury

- É. de la Clergerie is a member of the recruitment committee in Section 27 of University of Orléans.
- É. de la Clergerie was a member of the recruitment committee for INRIA CR2 at LORIA.
- É. de la Clergerie was a jury member for the PhD jury of Benoît Sagot [19].

9.4. Teaching

- É. de la Clergerie was invited to deliver a course on Parsing at ESSLLI (*European Summer School on Logic, Language and Information*) [18]

9.5. Committees

- Participation of É. de la Clergerie to the editorial board of French journal T.A.L. http://www.atala.org/rubrique.php3?id_rubrique=1.
- Participation of É. de la Clergerie to the program committees of HLT/NAACL'06, TALN'06, TAG+8, CSLP'06, and scientific committees of LREC'06. Participation to the program committees of coming TALN'07, ICLP'07 and IWPT'07. He has also reviewed papers for EACL'06.
- É. de la Clergerie has reviewed a proposal for the French program ANR MDCA "Masses de données - Connaissances Ambiantes".
- Participation of P. Boullier to the program committee of FG-2006 (*Formal Grammars*). He has also reviewed papers for EACL'06.
- B. Lang is vice-president of the SIL-CETRIL association for the economic development of the Soisson area (http://www.sil-cetril.org/article.php3?id_article=35).
- B. Lang has participated to the working group PIETA (Prospective de la propriété intellectuelle) of the Commissariat Général du Plan.

9.6. Participation to workshops, conferences, and invitations

- Participation of É. de la Clergerie to ISO TC37SC4 meetings (Paris, May; Beijing, China, August; and Brandeis University, MA, USA, October).
- B. Sagot has presented his work during different invited seminars, in Paris 7 (January), Labri/Bordeaux (February), LORIA/Nancy (April), DCU/Dublin (May), and Warsaw (June and August).
- É. de la Clergerie has presented Meta-Grammars at IRCS (University of Pennsylvania, October) and DCU/Dublin (November). He has presented meta-grammars and error mining at University of Santiago and University of Vigo (Spain, March).
- Participation with presentations of É. de la Clergerie at TALN'06 (*Traitement Automatique des Langues Naturelle*, Leuven) [31], ACL/COLING'06 (Sydney) [30]. Participation at TAG+8 (Sydney) and CSLP'06 (Sydney).
- É. de la Clergerie was invited to a 2-day working meeting at I3S (Univ. of Nice, September).
- Participation with presentations of B. Sagot at LREC'06 (Genova) [28], [25], TALN'06 (Leuven) [33], DLTAF'06 (ACFAS congress, Montreal) [29], Lexicon and Grammar conference (Palermo) [26]. B. Sagot also coordinates the seminar TALaNa.
- Participation and contribution of B. Lang to several meetings on the potential of Free Software, and on économic, legal and political issues.

10. Bibliography

Major publications by the team in recent years

[1] M. A. ALONSO PARDO, É. VILLEMONTE DE LA CLERGERIE, V. J. DIAZ, M. VILARES. *New Developments in Parsing Technology*, Text, Speech and Language Technology, John Carroll and Giorgio Satta and Harry Bunt (eds.), revised notes of a paper for IWPT2000, vol. 23, chap. Relating Tabular Parsing Algorithms for LIG and TAG, Kluwer Academic Publishers, 2004, p. 157–184, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/KAP_chapter8.Giorgio_John.pdf.

[2] P. BOULLIER. *A Cubic Time Extension of Context-Free Grammars*, in "Grammars", vol. 3, n^{o} 23, 2000.

[3] P. BOULLIER. *On TAG Parsing*, in "Traitement Automatique des Langues (T.A.L.)", issued June 2001, vol. 41, n^{o} 3, 2000, p. 111-131.

[4] P. BOULLIER. *Counting with Range Concatenation Grammars*, in "Theoretical Computer Science", vol. 293, 2003, p. 391–416.

[5] P. BOULLIER. *Guided EArley Parsing*, in "Proc. of the 8th International Workshop on Parsing Technologies (IWPT 03), Nancy, France", April 2003, p. 43–54, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Pierre.Boullier/earley_final.pdf.

[6] P. BOULLIER. *New Developments in Parsing Technology*, Text, Speech and Language Technology, John Carroll, Giorgio Satta and Harry Bunt (eds.), vol. 23, chap. Range Concatenation Grammars, Kluwer Academic Publishers, 2004, p. 269–289, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/KAP_chapter12.Giorgio_John.pdf.

[7] P. BOULLIER, L. CLÉMENT, B. SAGOT, É. VILLEMONTE DE LA CLERGERIE. *« Simple comme EASy :-) »*, in "Proceedings of TALN'05 EASy Workshop (poster), Dourdan, France", ATALA, June 2005, p. 57–60, http://atoll.inria.fr/~sagot/pub/TALN05easyworkshop.pdf.

[8] L. CLÉMENT, A. KINYON. *Generating parallel multilingual LFG-TAG grammars from a MetaGrammar*, in "Proc. of ACL'03", 2003.

[9] B. LANG. *Towards a Uniform Formal Framework for Parsing*, M. Tomita (ed.), Kluwer Academic Publishers, 1991, p. 153-171, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Bernard.Lang/framework.ps.Z.

[10] B. SAGOT, P. BOULLIER. *From raw corpus to word lattices: robust pre-parsing processing*, in "proc. of the 2nd Language & Technology Conference (LT'05), Poznan, Poland", Selected for potential journal publication, April 2005, p. 348–351, http://atoll.inria.fr/~sagot/pub/LTC05.pdf.

[11] B. SAGOT, L. CLÉMENT, É. VILLEMONTE DE LA CLERGERIE, P. BOULLIER. *Vers un méta-lexique pour le français : architecture, acquisition, utilisation*, Journée d'étude de l'ATALA sur l'Interface lexique-grammaire et lexiques syntaxiques et sémantiques, March 2005, http://atoll.inria.fr/~sagot/pub/JourneeATALA.pdf.

[12] B. SAGOT. *Automatic acquisition of a Slovak lexicon from a raw corpus*, in "Lecture Notes in Artificial Intelligence 3658 (© Springer-Verlag), Proceedings of TSD'05, Karlovy Vary, Czech Republic", September 2005, p. 156–163, http://atoll.inria.fr/~sagot/pub/TSD05.pdf.

[13] F. THOMASSET, É. VILLEMONTE DE LA CLERGERIE. *Comment obtenir plus des Méta-Grammaires*, in "Proc. of TALN'05, Dourdan, France", ATALA, June 2005, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/mg05.pdf.

[14] É. VILLEMONTE DE LA CLERGERIE. *Parsing Mildly Context-Sensitive Languages with Thread Automata*, in "Proc. of COLING'02", August 2002, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/COLING02.pdf.

[15] É. VILLEMONTE DE LA CLERGERIE. *DyALog: a Tabular Logic Programming based environment for NLP*, in "Proceedings of 2nd International Workshop on Constraint Solving and Language Processing (CSLP'05), Barcelona, Spain", October 2005, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/CSLP05.pdf.

[16] É. VILLEMONTE DE LA CLERGERIE. *Automates à Piles et Programmation Dynamique. DyALog : Une application à la programmation en Logique*, Ph. D. Thesis, Université Paris 7, 1993.

[17] É. VILLEMONTE DE LA CLERGERIE, M. A. ALONSO PARDO. *A tabular interpretation of a class of 2-Stack Automata*, in "Proc. of ACL/COLING'98", August 1998, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/SD2SA.ps.gz.

Year Publications

Books and Monographs

[18] É. VILLEMONTE DE LA CLERGERIE. *Designing tabular parsers for various syntactic formalisms*, Tutorial delivered at the 18th European Summer School in Logic, language and information (ESSLLI'06), July-August 2006, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/ESSLLI06.pdf, The association for Logic, Language and Information (FOLLI).

Doctoral dissertations and Habilitation theses

[19] B. SAGOT. *Analyse automatique du français: lexiques, formalismes, analyseurs*, PhD thesis, supervized by Laurence Danlos (Lattice, Université Paris 7) and co-supervized by Éric de La Clergerie (Atoll, INRIA Rocquencourt), Ph. D. Thesis, Université Paris 7, April 2006.

Articles in refereed journals and book chapters

[20] A. NASR. *Grammaires de dépendances génératives probabilistes. Modèle théorique et application à un corpus arboré du français*, in "Traitement Automatique des Langues", vol. 46, n^{o} 1, 2006, p. 115-153.

[21] A. NASR, O. RAMBOW. *Non-lexical chart parsing for TAG*, chap. Complexity of Lexical Descriptions and its Relevance to Natural Language Processing: A Supertagging Approach, MIT Press, 2006.

[22] F. ROLE, G. ROUSSE. *Construction incrémentale d'une ontologie par analyse du texte et de la structure des documents*, in "Document numérique", vol. 9, n^{o} 1, 2006, p. 77–92.

[23] B. SAGOT, L. DANLOS. *Améliorer un lexique syntaxique à l'aide des tables du lexique-grammaire – Constructions impersonnelles et expressions verbales figées*, in "Cahiers du Cental", submitted. Revised version for post publication of DLTAF 2006, 2006.

Publications in Conferences and Workshops

[24] F. BARTHÉLEMY. *Un analyseur morphologique multi-niveaux utilisant la jointure*, in "Proc. of TALN'06", 2006.

[25] P. BOULLIER, B. SAGOT. *Efficient parsing of large corpora with a deep LFG parser*, in "Proc. of LREC'06", 2006, http://atoll.inria.fr/~sagot/pub/LREC06a.pdf.

[26] L. DANLOS, B. SAGOT, S. SALMON-ALT. *French frozen verbal expressions: from lexicon-grammar tables to NLP applications*, in "Actes du Colloque Lexique et Grammaire 2006", 2006, http://atoll.inria.fr/~sagot/pub/lexgram06.pdf.

[27] M. FERNANDEZ, É. VILLEMONTE DE LA CLERGERIE, M. VILARES. *From text to knowledge*, in "Proc. of EUROCAST'07 (Eleven international conference on Computer Aided Systems theory)", to appear, 2006.

[28] B. SAGOT, L. CLÉMENT, É. VILLEMONTE DE LA CLERGERIE, P. BOULLIER. *The Lefff 2 syntactic lexicon for French: architecture, acquisition, use*, in "Proc. of LREC'06", 2006, http://atoll.inria.fr/~sagot/pub/LREC06b.pdf.

[29] B. SAGOT, L. DANLOS. *Améliorer un lexique syntaxique à l'aide des tables du lexique-grammaire*, in "Actes du colloque DLTAF 2006 (Description Linguistique pour le Traitement Automatique du Français) du congres de l'ACFAS, Montréal, Canada", 2006.

[30] B. SAGOT, É. VILLEMONTE DE LA CLERGERIE. *Error Mining in Parsing Results*, in "Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia", Association for Computational Linguistics, July 2006, p. 329–336, http://www.aclweb.org/anthology/P/P06/P06-1042.

[31] B. SAGOT, É. VILLEMONTE DE LA CLERGERIE. *Trouver le coupable : Fouille d'erreurs sur des sorties d'analyseurs syntaxiques*, in "Proc. of TALN'06", Prix du meilleur papier, 2006, p. 287-296, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/TALN06.pdf.

[32] D. SEDDAH, B. SAGOT. *Modeling and Analysis of Elliptic Coordination by Dynamic Exploitation of Derivation Forests in LTAG parsing*, in "Proceedings of TAG+8, Sydney, Australia", July 2006, p. 147-152.

[33] D. SEDDAH, B. SAGOT. *Modélisation et analyse des coordinations elliptiques via l'exploitation dynamique des forets de dérivation*, in "Proc. of TALN'06 (poster)", 2006, p. 609-618, http://atoll.inria.fr/~sagot/pub/TALN06b.pdf.

Miscellaneous

[34] J. MARTIN. *Mieux comprendre les Méta-Grammaires*, Technical report, Université Paris 6, September 2006, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/martin-DEA06.pdf.

[35] L. NICOLAS. *Fouille d'erreurs en analyse syntaxique*, Technical report, University of Nice, Master Recherche PLMT, September 2006, ftp://ftp.inria.fr/INRIA/Projects/Atoll/Eric.Clergerie/nicolas-DEA06.pdf.

[36] É. VILLEMONTE DE LA CLERGERIE. *ATOLL: Software Tools for Natural Language Processing*, Slides presented at Univ. of Vigo, Spain, March 2006.

[37] É. VILLEMONTE DE LA CLERGERIE. *From Meta-Grammars to Factorized grammars*, Slides presented at Univ. of La Coruña, Spain, March 2006.

[38] É. VILLEMONTE DE LA CLERGERIE. *Mining errors in large corpus parsing output*, Slides presented at Univ. of Vigo, Spain, March 2006.

References in notes

[39] M.-H. CANDITO. *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées*, Ph. D. Thesis, Université Paris 7, January 1999.

[40] B. CARPENTER. *The Logic of Typed Feature Structures with Applications to Unification Grammars, Logic Programs and Constraint Resolution*, n^{o} ISBN 0-521-41932, Cambridge University Press, 1992.

[41] S. EARLEY. *An Efficient Context-Free Parsing Algorithm*, in "Communications ACM 13(2)", ACM, 1970, p. 94-102.

[42] R. M. KAPLAN, J. BRESNAN. *Lexical-Functional Grammar: A formal system for grammatical representation*, in "The Mental Representation of Grammatical Relations, Cambridge, MA", J. BRESNAN (editor). , Reprinted in Mary Dalrymple, Ronald M. Kaplan, John Maxwell, and Annie Zaenen, eds., Formal Issues in Lexical-Functional Grammar, 29-130. Stanford: Center for the Study of Language and Information. 1995., The MIT Press, 1982, p. 173-281.

[43] G. A. KIRAZ. *Computational Nonlinear Morphology*, Cambridge University Press, 2001.

[44] A. NASR. *Analyse syntaxique probabiliste pour grammaires de dépendances extraites automatiquement*, décembre 2004, Habilitation à diriger des recherches, Université Paris 7.

[45] A. NASR, O. RAMBOW. *A Simple String-Rewriting Formalism for Dependency Grammar*, in "Workshop on Recent Advances in DependencyGrammar, International Conference on ComputationalLinguistics (Coling), Geneva, Switzerland", 2004, p. 25-32.

[46] A. NASR, O. RAMBOW. *Parsing with Lexicalized Probabilistic Recursive Transition Networks*, in "Finite-State Methods and Natural Language Processing 2005, Helsinki, Finland", 2005, p. 145-155.

[47] F. PEREIRA, D. WARREN. *Parsing as Deduction*, in "Proc. of the 21st Annual Meeting of the Association for Computationnal Linguistic, Cambridge (Massachussetts)", 1983, p. 137-144.

[48] C. POLLARD, I. A. SAG. *Head-Driven Phrase Structure Grammar*, University of Chicago Press, Chicago, 1994.

[49] D. WEIR. *Linear context-free rewriting systems and deterministic tree-walking transducers*, in "Proc. of ACL'92", 1992.