Fips, a "Deep" Linguistic Multilingual Parser

Eric Wehrli
LATL-Dept. of Linguistics
University of Geneva
Eric.Wehrli@lettres.unige.ch

Abstract

The development of robust "deep" linguistic parsers is known to be a difficult task. Few such systems can claim to satisfy the needs of large-scale NLP applications in terms of robustness, efficiency, granularity or precision. Adapting such systems to more than one language makes the task even more challenging.

This paper describes some of the properties of Fips, a multilingual parsing system that has been for a number of years (and still is) under development at LATL. Based on Chomsky's generative grammar for its grammatical aspects, and on object-oriented (OO) sofware engineering techniques for its implementation, Fips is designed to efficiently parse the four Swiss "national" languages (German, French, Italian and English) to which we also added Spanish and (more recently) Greek.

1 Introduction

This papers describes the Fips project, which aims at developing a robust, multilingual "deep" linguistic parsing system efficient enough for a wide-range of NLP applications. The system is currently available for six languages (English, French, German, Italian, Spanish and Greek), and has been extensively used for terminology extraction (Seretan & Wehrli, 2006), as well as for terminology assistance and translation (Wehrli, 2004, 2006).

This paper is organized as follows. The next section gives an overview of the Fips parser, describing some of its linguistic properties and its main processes. In section 3, we present the object-oriented design adopted for the project. Section 4 discusses some cases of cross-linguistic syntactic variation. Finally, section 5 provides some details about the results and presents an evaluation of the parser for the six languages.

2 The Fips parser

Fips is a robust "deep" linguistic parser which assigns to an input sentence an enriched S-structure type of representation, along with a predicate-argument representation. Fips can also be used as a tagger, outputing for each word of a given sentence a POS-tag and optionally the grammatical function (associated to the first word of a constituent), the base form (citation form) of the word, and whether a word is part of an expression or a collocation.

As an illustation, figure 1 shows the enriched structure representation and figure 2 the POS-tags returned by Fips for sentence (1). Notation is explained below.

(1) The record she broke was very old.

$[_{\texttt{TP}}$ $[_{\texttt{DP}}$ the $[_{\texttt{NP}}$ record$_i$ $[_{\texttt{CP}}$ $[_{\texttt{DP}}$ e$]_i$ $[_{\texttt{TP}}$ $[_{\texttt{DP}}$ she] broke $[_{\texttt{DP}}$ e$]_i$]]]]] was $[_{\texttt{FP}}$ $[_{\texttt{AP}}$ $[_{\texttt{Adv}}$ very] old]]]

Figure 1: Enriched S-Structure representation for sentence (1)

The linguistic assumptions used in this project correspond roughly to a free-adaptation of Chomsky's generative linguistics, borrowing concepts from the *Minimalist* model (Chomsky, 1995, 2004), from the *Simpler Syntax* model (Culicover & Jackendoff, 2005), as well as from *Lexical Functional Grammar* (Bresnan, 1982, 2001).

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 120–127,
Ann Arbor, June 2005.

word	tag	expression
the	DET-SIN	
record	NOM-SIN	break-record
she	PRO-PER-SIN-FEM	
broke	VER-PAS-3-SIN	break-record
was	VER-PAS-3-SIN	
very	ADV	
old	ADJ	

Figure 2: POS-tag output for sentence (1)

Roughly, the grammar is lexicalist, exploiting a rich lexicon which specifies, among others,

- the selectional properties of functional elements such as prepositions, auxiliaries, determiners, etc. For instance, the English auxiliary *have* selects a [+past participle] verbal projection. Similarly, in German, *werden* selects an infinitival verbal complement;
- arguments selected by predicative heads (nouns, verbs, adjectives);
- other syntactic or semantic features which might be relevant for syntactic processing such as [+pronominal] feature associated to certain verbs in French, Italian, German, etc., types and subtypes of adverbs, control properties of verbs selecting infinitival complements, and so on.

As shown in figure 1 above, the fundamental structures built by Fips all follow the same pattern, that is : LeftSubconstituents Head RightSubconstituents, which can be abbreviated as **L X R**, where **L** stands for the (possibly empty) list of left subconstituents, **X** for the (possibly empty) head of the phrase and **R** for the (possibly empty) list of right subconstituents. The possible values for **X** are the usual lexical categories **Adv**erb, **A**djective, **N**oun, **D**eterminer, **V**erb, **P**reposition, **C**omplementizer, **Inter**jection. To this list we add the functional category **T**ense, which is the head of a sentence (TP), as well as **F**unctional, used to represent predicative objects headed either by an adjective, an adverb, a noun or a preposition.

Compared to current mainstream Chomskyan representations, Fips constituent structures are relatively flat and make a rather parsimonious use of functional projections. They do, however, contain empty categories, either to represent empty subjects, for instance in infinitival complements, represented as sentences with a (usually lexically unrealized) subject. Empty categories are also used to represent "traces" of extraposed constituents, as in *wh*-constructions, where a chain of coindexed constituents is computed, headed by the extraposed element and footed by its "trace", an empty constituent in argument or adjunct position. An example of such chain is given in figure 1, where the noun *record* is first coindexed with the (lexically unrealized) relative pronoun in the specifier position of the CP constituent, which is itself related to the empty constituent $[_{\mathrm{DP}}\ e]_i$ in the canonical direct object position of the verb form *broke*.

Although quite complex, the computation of such chains brings many benefits in terms of quality and accuracy of the analysis. One clear example is provided by the identification of collocation, as exemplified in example (1) with the collocation *break-record*. In that sentence, the two terms of the collocation do not occur in the expected order and do not even occur in the same clause, since *record* is the subject of the main clause, while *broke* is in the relative clause. However, as the structure give in fig. 1 shows, the presence of the "trace" of *record* in the direct object position of the verb form *broke* makes the identification of the collocation rather simple, and fig. 2 confirms that Fips has indeed recognized the collocation.

The grammar itself consists of both rules and processes. Rules specify the attachment of constituents, thus determining, at least for the main part, the constituent structure associated to a sentence. The grammatical processes, which roughly correspond to some of the earlier transformation rules of Generative Grammar, are primarily responsible for tasks such as:

- filling up the argument table associated with predicative elements (mostly verbs);
- chain formation, ie. establishing a link between an extraposed element, such as a *wh*-element and an empty category in an argument or adjunct canonical position;
- modifications of the argument structure of predicates (adding, deleting, modifying arguments), as is necessary to account for passive or Romance causative constructions;
- coordination or enumeration structures.

In all such cases, the claim is that a procedural account is simpler than a rule-based description, leading furthermore to a more efficient implementation.

3 Object-oriented design

The computational model adopted for the Fips project relies on object-oriented (OO) concepts (see, for instance, Mössenböck, 1995). An abstract model is assumed both for objects and for their associated procedures (usually called "methods" in OO-jargon) – roughly corresponding to the "universal" linguistic level – from which language-specific objects and procedures are derived. In other words, linguistic objects are defined as abstract data types, whose implementation can vary from language to language. Such variation is handled by the type extension feature provided by OO-models when the variation concerns data structures or by the procedure redefinition feature when variation concerns a process.

Fips relies on three main objects:

- lexical units (*LexicalItem*), which correspond to the "words" of a language, as they appear in the lexical database;
- syntactic projections (*Projection*), which are the syntactic constituents;
- items (*Item*), which correspond to an analysis (partial or complete) – since the parser uses a parallel strategy, many items are maintained throughout the parsing process.

The main procedures (methods) associated to those objects are *Project*, *Merge* and *Move*, corresponding to the operation of projection, combination and movement, respectively. The following subsections will briefly discuss them in turn.

3.1 Project

The projection mechanism creates a syntactic constituent (an object of type *Projection* in our model), either on the basis of a lexical object, or on the basis of another syntactic constituent. For instance, any lexical item, as computed and retrieved from the lexical database by the lexical analysis is projected into a syntactic constituent, with the lexical item as its head. Thus, given *lex*, a lexical item, *lex.Project(p)* creates a syntactic projection *p* headed by *lex*, as in example (2):

(2)a. chat $\longrightarrow$ [$_{\mathtt{NP}}$ chat]

b. eine $\longrightarrow$ [$_{\mathtt{DP}}$ eine]

c. with $\longrightarrow$ [$_{\mathtt{PP}}$ with]

A more powerful variant of the projection mechanism, called metaprojection, can create richer syntactic constituents based either on specific lexical items or on other syntactic projections. For instance, we consider pronouns to be nominal-type lexical elements which project to a DP level. Similarly, verbs are taken as sentence heads, and will therefore give rise to richer syntactic constituents, as illustrated in the following examples:

(3)a. pronouns
[$_{\mathtt{DP}}$ [$_{\mathtt{NP}}$ toi]]

b. mangeras ("will-eat")
[$_{\mathtt{TP}}$ mangeras$_i$ [$_{\mathtt{VP}}$ e$_i$]]

c. reads
[$_{\mathtt{TP}}$ [$_{\mathtt{VP}}$ reads]]

d. regnet ("rains")
[$_{\mathtt{CP}}$ regnet$_i$ [$_{\mathtt{TP}}$ [$_{\mathtt{VP}}$ e$_i$]]]

Notice that the position of the tensed verb is different in the the structures given in (3b,c,d). We assume that tensed verbs in French (and more generally in Romance) "move" to the head of TP, as shown in (3b), while such movement does not occur in English (3c). An even more drastic example of metaprojection occurs in German, where we assume that a whole clause structure is projected on the basis of a tensed verb (in matrix clause), as illustrated in (3d).

3.2 Merge

Merge is the fundamental combination mechanism in our parsing model. Each time the parser reads a word, it is first transformed into a syntactic constituent, a projection, as we have just seen. The projection, in turn, must now be combined (merged) with (partial or complete) constituents in its immediate left context. Two scenarios are considered, corresponding to left attachment and to right attachment. Left attachment occurs when the projection in the left context of the new projection can be attached as a left subconstituent of the new projection. Right attachment corresponds to the situation where the new projection can be attached as a right subconstituent of the projection in its left context. In fact, to be more accurate, the incoming projection can attach as a subconstituent not just of the projection in its left context, but to any active node of it, as illustrated in Figure 3 below:

Merge must be validated either by lexical properties such as selectional features or by general properties (adverbs, adjuncts, parentheticals can relatively freely modify projections).

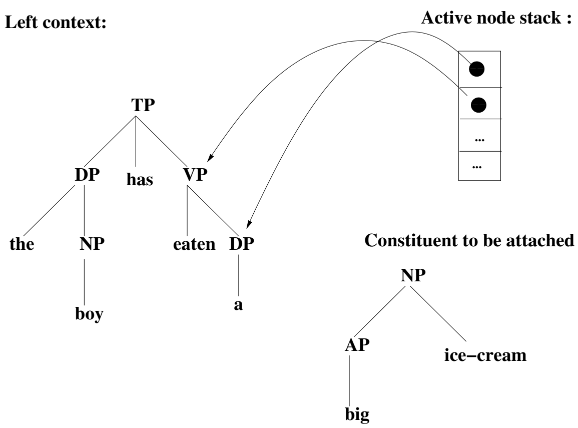

Figure 3: Active node stack

3.3 Move

Although the general architecture of surface structures results from the combination of projection and merge operations, an additional mechanism is necessary to satisfy well-formedness conditions such as thematic assignment. As mentioned earlier, such a mechanism reveals quite useful for the collocation identification process (cf. fig. 2). This mechanism handles extraposed elements and link them to empty constituents in canonical positions, thereby creating a chain between the base (canonical) position and the surface (extraposed) position of the *moved* constituent. To take another simple example, let us consider the interrogative sentence given in (4a) and the (slightly simplified) associated structure in (4b):

(4)a. who did you invite ?

b. [$_{CP}$ [$_{DP}$ who]$_i$ did$_j$ [$_{TP}$ [$_{DP}$ you] e$_j$ [$_{VP}$ invite [$_{DP}$ e]$_i$]]]

The chain mechanism functions as follows: as the parser encounters a *wh*-word in an extraposed position, it stores it in a stack associated with its governing category (typically the CP projection which dominates it). As the parse proceeds, the stack is transferred along the right edge of the structure, provided it does violate island conditions (cf. Ross, 1967). Whenever a predicative element is added to the structure, an attempt is made to complete the chain, ie. to interpret the projection on top of the stack with respect to the predicate. If successful, an empty category coindexed with the extraposed projection is inserted into the structure and the projection is removed from the stack. At the end of the parse, items containing unresolved chains are heavily penalized.

3.4 A parsing example

To illustrate the parsing procedure and the interaction of the 3 mechanisms described above, consider the following simple sentence in French.

(5)a. Paul mange une pomme.
'Paul eats an apple'

b. [$_{TP}$ [$_{DP}$ [$_{NP}$ Paul]] mange$_i$ [$_{VP}$ e $_i$ [$_{DP}$ une [$_{NP}$ pomme]]]]

Step 1 the parser reads "Paul" and metaprojects a DP structure [$_{DP}$ [$_{NP}$ Paul]].

Step 2 the parser reads "manges" and metaprojects a TP-VP structure [$_{TP}$ mange$_i$ [$_{VP}$ e$_i$]]. A merge operation is possible with the preceding DP structure, which yields [$_{TP}$ [$_{DP}$ [$_{NP}$ Paul]] mange$_i$ [$_{VP}$ e$_i$]].

Step 3 the parser reads the determiner "une" and creates a DP structure [$_{DP}$ une]. A merge operation is possible with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node [$_{TP}$ [$_{DP}$ [$_{NP}$ Paul]] mange$_i$ [$_{VP}$ e$_i$ [$_{DP}$ une]]].

Step 4 the parser reads the noun "pomme", creates an NP structure [$_{NP}$ pomme], and attach it (merge operation) as a right constituent of the DP structure in the TP structure, which yields the complete structure (5b).

3.5 The grammar

Merge operations are constrained by various mostly language-specific conditions which can be described by means of rules. Those rules are stated in a pseudo formalism which attempts to be both intuitive for linguists and relatively straightforward to code. The conditions associated to the rules take the form of boolean functions, as described in the examples (6) for left attachments and in the examples (7) for right attachments, where **a** and **b** refer, respectively, to the first and to the second constituent of a merge operation.

(6)a. AP + NP
a.HasFeat(prenominal)
a.AgreeWith(b, {gender, number})

b. DP + NP
a.HasSelectionFeat(nCompl)
a.AgreeWith(b,{gender, number, case})

Rule 6a specifies that an adjective projection structure (an AP constituent) can (left-)merge with a noun projection structure (an *NP* constituent) under the two conditions (i) that the first constituent (the adjective) bears the feature `prenominal` and (ii) that both constituents agree in number and gender. This rule, which is part of our French grammar, will allow for *petit animal* ("small animal"), but not *préhistorique animal* ("prehistorical animal"), since the adjective *préhistorique* does not bear the feature `[+prenominal]`, nor *petit animaux* ("small animals"), since *petit* is singular while *animaux* is plural and hence both do not agree in number.

Rule (6b) is taken from our German grammar. It states that a common noun can be (right-)attached to a determiner phrase, under the conditions (i) that the head of the DP bears the selectional feature `[+Ncomplement]` (ie. the determiner selects a noun), and (ii) the determiner and the noun agree in gender, number and case.

3.6 Procedural grammar

One of the original features of the Fips parser is its procedural approach to several grammatical properties. In addition to the chain mechanism described in the previous section, the procedural approach also concerns the treatment of passive and other restructuring constructions, as well as coordination. The next two paragraphs briefly sketch our treatment of passive and coordination constructions.

3.6.1 passives

Along with many linguists or various persuasions, we assume that the fundamental property of passives is the elimination (or demotion) of the subject argument of a predicate. Based on that assumption, our treatment is essentially a case of argument-structure modification: demotion of the subject argument to an optional "*by*-phrase" argument, promotion of the direct object argument to the subject argument slot. In our implementation, the treatment of passives takes the form of a grammar rule specifying the attachment of a [+past participle] verbal projection as complement of the passive auxiliary. This attachment triggers the restructuring process described above[1].

3.6.2 coordination

Coordinate structures constitute a well-known problem for both theoretical and computational linguistics. For the latter, coordination is problematic because it is a major source of non-determinism. Given the fact that such structures are extremely common in both speech and writing, it is therefore mandatory for NLP systems to handle them efficiently. Our treatment of coordination is based on the following assumptions:

- Coordination can affect any pair of like constituents;
- coordinate structures do not strictly obey the $\overline{\text{X}}$ schema. They have the following structure: $[_{\texttt{XP}}\ [_{\texttt{ConjP}}$ XP Conj XP]], where X takes its value in the set of lexical categories augmented by T and F (see section 2 above), and CONJ is a coordination conjunction (eg. *and*, *or*, *but*, etc.).

The coordination procedure is triggered by the presence of a conjunction. All the nodes on the right edge of the constituent in its immediate left context are considered potential candidates for the coordination structure. A metaprojection creates a coordinate projection, in which the node on the right edge is the left subconstituent of the conjunction. The set of such projections is quickly filtered out by further incoming material.

To illustrate our treatment of coordinate structures, in particular the type of structure we assume (slightly simplified in the (8) sentences) as well as the potential ambiguity of coordination, consider the following simple English examples.

(7)a. the old men and women

b. $[_{\texttt{DP}}\ [_{\texttt{ConjP}}\ [_{\texttt{DP}}$ the $[_{\texttt{NP}}\ [_{\texttt{AP}}$ old] men]] and] $[_{\texttt{DP}}\ [_{\texttt{NP}}$ women]]]

c. $[_{\texttt{DP}}$ the $[_{\texttt{NP}}\ [_{\texttt{AP}}$ old] $[_{\texttt{ConjP}}\ [_{\texttt{NP}}$ men] and $[_{\texttt{NP}}$ women]]]]

[1]The same restructuring process applies to particial structures, as in John left the room, **followed** by his dog.

d. [$_{DP}$ the [$_{NP}$ [$_{ConjP}$ [$_{NP}$ [$_{A}$ old] men]] and] [$_{NP}$ women]]

(8)a. John believes Bill and Mary will be to blame.

b. John believes [$_{TP}$ [$_{DP}$ Bill and Mary] will be to blame]

c. [$_{TP}$ John believes Bill] and [$_{TP}$ Mary will be to blame]

4 Examples of cross-linguistic variation

In the Fips system, language variation occurs not only at the level of the grammar, as expected, but also at the level of the associated procedures. Consider for example, the case of the argument checking procedure. Whereas a preverbal DP can be interpreted as the subject of a verb if it agrees with it (number, person) in languages such as French or English (as well as other so-called "configurational languages"), the same criteria would not hold for case-marked languages, such as German or Modern Greek. In those languages, subjects can essentially occur anywhere in the sentence but must be marked `[+nominative]` and of course agree with the verb (number, person)[2]. Relatively similar at an abstract level, the argument checking procedure must be "tuned" for each individual language.

Our second example of cross-linguistic variation concerns clitic pronouns. The relevant data structures (objects) and interpretation procedures (methods) to handle clitics are defined at an abstract level. Specific languages (ie. Spanish, Italian, French, Greek, etc.) inherit those objects and methods, which they can further specialize according to language-specific properties and constraints. The general mechanism to handle clitics comprises two distinct steps: attachment and interpretation[3]. As a clitic is read (as an independent word or as an orthographically attached affix), it is attached to the head of the verb form which follows it (proclitic) or which precedes it (enclitic). Since this verbal head is not necessarily the one with respect to which the clitic pronoun can be interpreted (it might be an auxiliary, for instance), a temporary data structure is used to store clitics until the parser has identified the main predicate of the sentence[4]. Only then can the interpretation process start. All the clitics in the temporary data structure must be interpreted either as argument or as adjunct of the verb[5]. The examples below illustrate our analysis of clitics, applied to Italian (9), French (10) and Spanish (11). The Italian and French examples display proclitics (pre-verbal clitics), while the Spanish example is a case of enclitics (post-verbal clitics). Notice also that in Italian and Spanish we have clitic clusters (two clitics concatenated in one orthographical word), and in the Spanish example, the cluster is itself concatenated to the verb. In all three examples, the clitic pronouns have be properly analyzed, ie. interpreted as arguments of the verb. This is expressed in the resulting structures by the chains connecting a pronoun and an empty category in postverbal position. As in the *wh*-chains discussed earlier, all the elements are coindexed.

(9)a. Glielo ho dato. ("I have given it to him")

b. [$_{TP}$ [$_{DP}$ e] gli$_i$-lo$_j$ ho [$_{VP}$ dato [$_{PP}$ e$_i$] [$_{DP}$ e$_j$]]]

(10)a. Paul le lui a donné. ("Paul has given it to him")

b. [$_{TP}$ [$_{DP}$ Paul] le$_i$ lui$_j$ a [$_{VP}$ donné [$_{DP}$ e$_i$] [$_{PP}$ e$_j$]]]

(11)a. Dámmelo. ("Give it to me")

b. [$_{TP}$ [$_{DP}$ e] da$_i$-me$_j$-lo$_k$ [$_{VP}$ e$_i$ [$_{PP}$ e$_j$] [$_{DP}$ e$_k$]]]

Although very similar in their fundamental behavior, clitics across Romance languages are nevertheless too different to be handled by exactly the same mechanism. Furthermore, even if such mechanism could be implemented, chances are that it would prove insufficient or inadequate in some ways to handle an additional Romance language such as Romanian or Portuguese. Our approach, based on a general abstract mechanism, which can be specialized to suit the specific properties of each language seems therefore more appropriate.

[2]We assume that German (and Modern Greek) are so-called scrambling languages with an unmarked basic word order (cf. Haider and Rosengren, 1998, Hinterhölzl, 2006).

[3]From now on, the discussion will only focus on Romance clitics.

[4]This temporary structure is also used to check the well-formedness of clitic sequences.

[5]For the sake of simplicity, we will leave aside a few more complex cases, such as French clitic "en" corresponding to complements of the direct object of the main verb (*Paul* **en** *connaît la raison* "Paul knows the reason of it") or so-called "long-distance" clitics in Italian or Spanish restructuration constructions.

5 Results and evaluation

To date, the Fips multilingual parser has been developed for 6 languages (English, French, German, Italian, Spanish and Greek). Other languages have been very partially treated, such as Romanian, Russian, Polish and Romansch Sursilvan.

A significant effort has been made at the lexical level, qualitatively and quantitatively. The table in figure 4 below shows the curren approximate size of each lexicon.

language	lexemes	words	collocations
anglais	54'000	90'000	5'000
français	37'000	227'000	12'500
allemand	39'000	410'000	2'000
italien	31'000	220'000	2'500
espagnol	22'500	260'000	320
grec	12'000	90'000	225

Figure 4: Number of entries in the lexical database

At the grammar level, the coverage of the English and French grammar is quite satisfactory, Italian, Spanish and especially German still need improvements, while the Greek grammar is very partial.

Fips attempts to produce complete analyzes for input sentences. Since the parsing strategy is (pseudo-)parallel, many analyzes are produced and ranked according to preferences such as local vs. non-local attachments, argument vs. adjunct interpretation, presence vs. absence of a collocation, etc. When a complete analysis fails, the parser outputs a sequence of partial analyzes covering the whole sentence.

A comparative evaluation has been conducted to show how the various implementations of Fips compare with respect to a near identical corpus, the European Parliament corpus (cf. Koehn, 2005). We parsed approximately 1 million words in each of the six languages. The table given in figure 5 show the results:

The first line in table 5 show the size of each file in terms of symbols (word, punctuation, formatting symbol, etc.), approximately 1 million symbols for each file. The second line gives the number of unknown words, not counting words starting with an uppercase letter which are assumed to be proper nouns (given the fact that in German common nouns are capitalized, we did not leave aside capitalized unknown words for that language). The third line indicates the number of sentences approximately 40'000 for each file, slightly more for the German file. We can see that the average length of a sentence is roughly 20 to 25 symbols (slightly more for French). The fourth line shows the percentage of sentences for which Fips returned a complete analysis. The best score is obtained with English (71.95%), closely followed by French (70.01%). Greek is clearly behind with only about 31%, largely due to the fact that its grammar as well as its lexicon have received much less attention so far. We can observe a quite clear (and unsurprising) correlation between rich lexical coverage (English, French) and high number of complete analyzes.

Finally the last line shows the speed of the parser in terms of number of words per second. The mean speed of Fips is between 130 and 180 word/second. FipsGreek is somewhat faster, presumably because its grammar is less developed than the grammar of the other languages at this point. It came up as a surprise to see that FipsEnglish was clearly slower. The reason has probably to do with the high number of lexical ambiguities of the type N/V (e.g. lead, study, balance, need) which are likely to significantly increase the number of parallel (partial) analyzes.

6 Concluding remarks

Although the research described in this paper is by no means completed, it has already achieved several important goals. First of all, it has shown that "deep linguistic parsing" should not necessarily be equated with "inefficient parsing". Although clearly slower than shallow parsers, Fips is fast enough for such demanding tasks as translation or terminology extraction.

At the software level, the adopted design makes it possible to "plug" an additional language without any change or any recompilation of the system. It is sufficient to add the language-specific modules and lexical databases to have a fully functional parser for that language. Arguably the model has so far not been tested with languages belonging to widely distinct language types. In fact, it has only been applied to (a small set) of European languages. Future work will address that issue, and we are planning to extend our work towards Asian and Semitic languages.

language	German	English	Spanish	French	Greek	Italian
number of symbols	1082117	1046431	1041466	1144345	1045778	998871
unknown words	13569	879	6825	853	26529	3099
number of sentences	45880	40348	40576	38653	39812	37726
% of complete analyzes	48.04%	71.95%	56.87%	70.01%	30.99%	58.74%
speed (word/second)	138	82	127	133	243	182

Figure 5: Comparative evaluation of the parsers

Acknowledgement

Thanks to Luka Nerima, Christopher Laenzlinger, Gabriele Musillo and Antonio Leoni de León for various suggestions and comments on earlier versions of this paper. The research described here has been supported in part by a grant from the Swiss national science foundation (no 101412-103999).

7 References

Bresnan, J. (éd.), 1982. *The Mental Representation of Grammatical Relations*, Cambridge, Mass., MIT Press.

Bresnan, J., 2001. *Lexical Functional Syntax*, Oxford, Blackwell.

Chomsky, N. 1995. *The Minimalist Program*, Cambridge, Mass., MIT Press.

Chomsky, N. 2004. "Beyond Explanatory Adequacy", in A. Belletti (ed.) *The Cartography of Syntactic Structures*, Oxford, Oxford University Press.

Culicover, P. et R. Jackendoff, 2005. *Simpler Syntax*, Oxford, Oxford University Press.

Haider, H. and I. Rosengren 1998. "Scrambling" *Sprache und Pragmatik* 49, Lund University.

Hinterhölzl, R. 2006. *Scrambling, Remnant Movement and Restructuring in West Germanic*, Oxford, Oxford University Press.

Koehn, Ph., 2005. "Europarl: A Parallel Corpus for Statistical Machine Translation, MT Summit.

Mössenböck, H. 1995. *Object-Oriented Programming in Oberon-2*, New York, Springer.

Ross, .R. 1967. *Constraints on Variables in Syntax*, Ph.D. dissertation, MIT.

Seretan, V. et E. Wehrli, 2006. "Accurate collocation extraction using a multilingual parser" in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*, Sydney, 952-960.

Wehrli, E. 2004. "Traduction, traduction de mots, traduction de phrases", in B. Bel et I. Marlien (eds.), *Proceedings of TALN XI*, Fes, 483-491.

Wehrli, E. 2006. "TwicPen : Hand-held Scanner and Translation Software for non-Native Readers", in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*, Sydney.