



Verbal chunk extraction in French using limited resources

Gabriel G. Bès, Lionel Lamadon, François Trouilleux

► To cite this version:

Gabriel G. Bès, Lionel Lamadon, François Trouilleux. Verbal chunk extraction in French using limited resources. 2004. <hal-00002699>

HAL Id: hal-00002699

<https://hal.science/hal-00002699v1>

Preprint submitted on 26 Aug 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Verbal chunk extraction in French using limited resources *

Gabriel G. Bs

Lionel Lamadon

François Trouilleux

Universit Blaise-Pascal

34, avenue Carnot

F-63037 Clermont-Ferrand Cedex, France

Gabriel.Bes@univ-bpclermont.fr, Lionel.Lamadon@univ-bpclermont.fr,
trouilleux@lrl.univ-bpclermont.fr

August 26, 2004

Abstract

A way of extracting French verbal chunks, inflected and infinitive, is explored and tested on effective corpus. Declarative morphological and local grammar rules specifying chunks and some simple contextual structures are used, relying on limited lexical information and some simple heuristic/statistic properties obtained from restricted corpora. The specific goals, the architecture and the formalism of the system, the linguistic information on which it relies and the obtained results on effective corpus are presented.

1 Introduction

It is generally admitted that taggers or chunkers, either statistical or grounded on linguistic knowledge, will never reach perfect results ¹. However, it is also generally admitted that there are cases where, using only local contextual part-of-speech information and chiefly information coming from morphological items, deductions on categories or chunks can be made with utmost certainty ².

In this situation, the crucial question is: what results can be obtained with such and such information ? or, conversely, what information does one need to obtain this particular result ? We here try to address these issues with the presentation of a system that has as its immediate target the identification of verbal chunks — finite and infinitive — in French.

We will readily agree that this target is fairly limited, compared to that accomplished by general purpose taggers and chunkers. However, to dig into a problem, common practice is to split it into smaller ones: our goal is to identify as clearly as possible the possibilities and limits of different types of information with respect to specific structures, *i.e.*, for the matter, verbal chunks ³.

Given this issue, the general and strong requirement which underlies the specification of our verbal chunk extractor is: do the more with the least. That is, we want to maximally reduce the resources

*Thanks to Zulema Solana for her comments on a previous version of the paper, the responsibility of which is entirely ours.

¹Abney [1] gives the following example: “to correctly disambiguate *help* in *give John help_N* versus *let John help_V*, one apparently needs to parse the sentences, making reference to the differing subcategorization frames of *give* and *let*.”

²Tapanainen and Voutilainen [11] discuss this point and give the following example: “we can safely exclude a finite-verb reading if the previous word is an unambiguous determiner.”

³Our system belongs thus to partial parsing (cf. Manning & Schütze [8, p. 375]) but with verbal chunks as the target of the analysis and not the most currently practiced nominal phrases. The reason of our choice is the potentiality of inferences on the syntactic analysis of the whole sentence that can be driven from analyzed verbal chunks; on the importance of correctly tagging verbs, see Chanod & Tapanainen [5] and footnote 6.

necessary to put the system into use, either lexical information or tagged corpus, but keeping in mind that this must not hamper final results on effective texts. Our general underlying hypothesis is : with (i) a simple linguistic modelization, chiefly relying on morphological expressions in verbal, nominal and prepositional chunks, plus (ii) very limited lexical resources, plus (iii) heuristic statistical information on forms not covered by the previous (i) and (ii), it is possible to obtain results standardly considered as good, *i.e.* around 98% of recall and precision ⁴.

As a consequence, the system is centrally rule-based, but is statistically completed. The satisfaction of the general requirement on resource poverty settles the balance between the two types of contributions. We use as much as possible a simple linguistic modelization, which exempts from the use of large tagged corpora, and we complete it with very simple statistical information. People more or less agree on the respective advantages and disadvantages of corpus statistics vs. linguistic knowledge (see Chanod & Tapanainen [5] and Brill [4] which represent opposite orientations) and both approaches can or must – ordinarily, they are – be put to work together, the most important difference coming on the weight given to each one.

In general terms our system can be characterized as working in the opposite way of the transformational-based learning way of tagging (on this, see Brill [4], Charniak [6], Manning & Schtze [8]) ⁵.

It is known that the transformational way of tagging is organized in two steps. In the first step, a simple algorithm records for each word its most common part of speech in the training corpus. The tagging of new texts consists in simply assigning its most common tag to each word. The reported results of this level are 90% accuracy. In the second step, rules are proposed and evaluated in order to change the tags obtained in the first step into others intended to be more accurate. E.g. related to our issue, a rule can be:

Change a verbal tag into a nominal tag if the previous word is a non ambiguous determiner.

Transformations are not specified in terms of linguistic knowledge. Given $|PoS|$ the total number of parts of speech, there are $|PoS|^3$ possible transformational rules of the previous type. The necessary training data - *i.e.* tagged corpus of several hundred thousands words - are then put into use. The system tries each possible rule on the training data measuring the accuracy the rule induces. And Charniak reports:

Some rules make things worse, but many make things better, and we pick the rule that makes the accuracy the highest. Call this rule 1.

Next we ask, given we have already applied rule 1, which of the other rules does the best job of correcting the remaining mistakes. This is rule 2. We keep doing this until the rules have little effect. The result is an ordered list of rules, typically numbering 200 or so, to apply to new examples we want tagged.

In our system, structural rules (see below section 5) express the simple needed linguistic modelization. They are the counterparts to the transformational rules for tags changing, but they are not learned by some algorithm and they are not used for erasing or modifying already obtained results coming from a previous step. Our heuristic/statistical information is the analogous of statistical information calculated on the first step of the transformational-based learning strategy, with the big differences that it does not concern all words related to verbal chunks, but only the ones whose tags are not assigned by structural rules, and that our “training” corpus is extraordinary reduced compared with training corpora used in the statistical ways of doing things.

⁴Manning & Schtze [8, p. 371] report the range of 95% to 97% for accuracy numbers calculated over all words; Charniak [6] reports 97%; Habert & al. [7] recall the standard figures of 95% to 98%. Reported figures for French are 96.8% for a statistical tagger and 98.7% for a constrained based one with heuristic rules (see Chanod & Tapanainen [5]). In general it is difficult to state clearly revealing figures given the different parameters intervening in the tagging tasks.

⁵The system is thus, in *spirit* if not necessarily in expressive or covering power, in the line of Chanod & Tapanainen [5], the EngCG tagger [13], Tapanainen & Voutilainen [11], Tapanainen & Jrvinen [10].

In the following, section 2 specifies the targeted task of the system, section 3 presents the system architecture, section 4 describes linguistic resources and section 5 illustrates rules of the system. Section 6 is devoted to the analysis of results obtained by a detailed evaluation: we are not only interested by the obtention of global figures but also by a more detailed one, allowing to discriminate between the different types of parameters which contribute to global results. Particularly, we want to discriminate between results obtained thanks to the linguistic modelization and results obtained thanks to heuristic/statistical choices, to discriminate between missing resources and effective counter-examples, and we want also to be sure that “good” results do not necessarily come from a high or dominant percent of non ambiguous forms ⁶. The perspectives of the system are discussed in the final section 7.

2 Task Definition

In French, verbal chunks may include, in addition to a verb nucleus:

- auxiliary verbs, of two types as in *Jean a mang* or in *Jean est reparti*,
- clitic pronouns, as in *Jean le lui prend*,
- negation particles, as in *Jean ne parle pas*,
- adverbs, as in *Jean a rapidement mang*,
- occasionally, incidental phrases or clauses, as in *Jean a, par ailleurs, mang*,
- and, in infinitive verbal chunks, a preposition, as in *Il parle de le prendre*.

With the only exception of the negative particle *ne*, and of some reflexive forms, all the morpheme expressions in verbal chunks are ambiguous : *le* can be an article or a clitic pronoun, *lui* a nominative or a dative clitic pronoun, *en* a preposition or a clitic pronoun, *pas* a particle in negation constructions or a noun, etc. Furthermore, many verbal expressions are ambiguous: *note* can be a verb or a noun, *avoir* a noun or the infinitive of one of the two auxiliary forms, *savoir* a noun or an infinitive, *entre* a verb or a preposition, *inverse* a verb or an adjective, *lui* the past participle of the verb *lui*re or the clitic or the nominative pronoun, *soit* a verb or a conjunction, etc. Besides this, verbal forms used as auxiliaries can also be used as main verbs as in *Jean est triste* (*John is sad*) ou *Marie a la migraine* (*Mary has got a headache*). But, despite this superabundance of several different kinds of ambiguities, for a human reader, effective texts are never ambiguous with respect to verbal chunks.

One central challenge of verbal chunk extraction is thus to solve morphological and/or lexical and/or functional ambiguities. But we do not think that disambiguation of items in the whole sentence or, even, in the targetted structures, is a step which must necessarily be achieved before the obtention of chunks. In any case, we want to find out how far one can go with different kinds of limited information and combining disambiguation with chunk extraction ⁷.

⁶Manning & Schtze [8, p. 374–375] points that “some authors give accuracy for ambiguous words only in which the accuracy figures are of course below”. In the literature reporting results it is not always possible to find discriminated results for ambiguous and not ambiguous forms. Even in Chanod & Tapanainen [5] where actually some figures are discriminated, the ones on the overall performance of the reported system are not. The point is important from a methodological viewpoint; see Habert & al. [7] which points furthermore that tagging errors have quite different consequences on the syntactic analysis of the whole sentence depending on the kind of ambiguity: a tagging error on the noun/verb ambiguity will more significantly hamper the results on the syntactic sentence structure than a tagging error on the past participle/adjective ambiguity.

⁷Manning & Schtze [8, p. 374–375], observes that although the interest in tagging comes from the belief that many engineering applications to language processing must benefit from syntactically disambiguated texts, “it is surprising that there seem to be more papers on stand-alone tagging than to applying tagging to a task of immediate interest”. Our system aims at overcoming this criticism by combining tagging with chunk extraction.

3 System Architecture

The system consists in two modules: Smorph, which performs tokenization and morphological analysis (At-Mokhtar [2]), and Pasmio (Paulo et al. [9]), which, given the output of Smorph, delimits sentences and, in the present case, verbal chunks, based on “recomposition rules” similar to rewriting rules. Smorph and Pasmio are related by a script interface tagging unknown words by Smorph. In addition, the system is associated to an evaluation software (Trouilleux [12]) which can provide detailed evaluation results (cf. section 6)⁸.

Given a text file, Smorph outputs a list of tokens of the form:

```
surface_form [lemma, FVL]
```

where FVL is a feature/value list.

A Pasmio recomposition rule is of the form:

```
token1 ... tokenn -> token1' ... tokenn'
```

with the reading “a sequence of tokens matching the left-hand side of the rule is to be recomposed into the sequence of tokens specified in the right-hand side of the rule” (*i.e.* basically, the reading is the inverse of that of traditional rewriting rules). The Kleene + operator may be used in the left-hand side of the rule. In the new tokens produced by a rule, the `surface_form` and `lemma` elements may either be specified as specific strings of characters or as the concatenation of the `surface_form` and `lemma` elements of the input tokens, while the feature/value pairs are to be specified as strings of characters only. Examples of rules are given in section 5.

The data structure output by application of Pasmio rules is the same as that output by Smorph, allowing cyclic application: the rules are first applied to the output of Smorph, producing output 1, then applied again using output 1 as an input, producing output 2, then applied again, ... until the input is such that no rule applies.

The purpose of this paper is not so much to introduce a new chunking system, but rather to analyse the importance of different types of information in a chunking process related to verbal forms. We thus won't go into a detailed comparison of Smorph-Pasmio with existing systems.

4 Lexical Linguistic Information

The declarative source of Smorph is a lexicon in which the linguist declares lemmas and the set of feature/value pairs associated to each of them in terms of their inflected forms. It is possible to express in Smorph compound words, hyphenated, as in *remue-mnage* (*commotion*) or not, as in *pomme de terre* (*potatoe*).

Contrary to general practice, the basic choices with respect to our lexicon are:

- With the exception of verbal forms which are ambiguous as `fl` (inflected) or `pp` (past participle), ambiguity is not expressed by the multiplication of the same forms. E.g. there are not two forms *le*, one associated to a value `art[icle]` and the other to a value `cl[itic]`, nor two forms *note*, one associated to `n[oun]` and the other to `v[erb]`; instead there is a unique form *le* associated to the value `ocl` of the feature `AMB[iguity]`, which expresses the potential ambiguity of the clitic, and a unique form *note* associated to the value `ov` of the same feature, which expresses the potential ambiguity of the verb.
- With the only exceptions of plural first and second person verb forms, no flexional values are associated to morphological or lexical forms⁹.

⁸Thanks to Mourad Sahi for his contribution with a program to compile results obtained on a set of files.

⁹The value is needed in order to distinguish contextually a subject *nous*, as in *nous voulons* (*we want*) from a clitic one, as in *Il nous parle* (*He talks to us*).

- Lexical categories are drastically reduced. There is no noun category, and adjective entries are less than ten.

In its present state, as far as expressions occurring inside verbal chunks are concerned, our lexicon contains the following forms:

1. all the verbal forms of our “training” and test corpus ¹⁰,
2. all the morphological forms which can appear in verbal chunks: clitic pronouns, *ne*, *pas*, prepositions in infinitive chunks (*de*, etc.),
3. a subset of a restricted class of adverbs as *ainsi*, *d’ailleurs*, *alors*,
4. a subset of multiword frozen adverbial expressions which incorporate forms which, in isolation, can be verbs (e.g. *sans doute*, *n’importe qui*, where *doute* and *importe* alone can be verbs).

Items (1) and (2) are intended to be exhaustive, while this is not the case of items (3) and (4). Verbal forms are categorized as follows:

- inflected forms *f1* are either categorized as *ov* (potentially ambiguous forms), as *note*, or categorized *nv* (not potentially ambiguous forms), as *accepte*; infinitive forms *inf* are either categorized as *oin* (potentially ambiguous forms), as *devenir*, or categorized *nin* (not potentially ambiguous forms), as *assurer*,
- a subset of the potentially ambiguous inflected forms (category *ov*) is further categorized as having a statistically unlikely verb reading, (category *jv*, *j[amais] v[verbe]*, *never verb*), as *empire*,
- past participles, whether they are ambiguous with respect to nouns (e.g. *entre*) or not (e.g. *dormi*), belong to a unique category *pp*,
- verbal forms which are ambiguous as *f1* (inflected) or *pp* (past participle), e.g. *fait*, are associated in the lexicon to two different forms, one categorized as *f1* and the other as *pp*.

In addition to this categorization, as stated above, first and second person verbal forms (and only those) are associated to flexional information in the form of a person and number feature.

There are in all in the lexicon 1.731 verbal forms: 658 *f1*, 595 *pp*, 478 *inf*. With respect to ambiguity, the significant figures are: 429 (about 2/3) *f1* forms are *nv* while 229 are *ov*; 440 (92%) *inf* forms are *nin* while 38 are *oin* ¹¹.

In addition to forms occurring in verbal chunks, the lexicon also contains some prepositions, determiners which are not ambiguous with forms inside verbal chunks, as *un*, *des*, *ce...*, unambiguous possessives, as *mon*, *ses...*, a small set of adjectives (less than ten), a small set of pronouns, as *il*, *on...* and punctuation marks. The forms other than punctuation marks which are not verb forms and which can appear outside verbal chunks (this is not the case, e.g. of nominative or some reflexive clitics as *-il*, *me*, *se*), are all categorized as *auf* (*au[tre] f[orme]*, *other form*). There are in all 219 *auf* forms in the lexicon.

With so limited lexical resources it is obvious that not all token forms in texts will be associated by Smorph to some feature/value categorization. In fact, in both the “training” and test corpus a little more than 1/3 of the forms are not recognized by Smorph and are associated by the script interface to [*au*, *auf*] feature values (*au* being a feature value not used in the lexicon). Overall, 2/3 of the tokens are associated to the most frequent feature value *auf*.

¹⁰In order to examine the minimal lexical information (in terms of categories) required for our proposed task, before running the system on a given corpus, we make sure that to all the verbal and potentially verbal forms in the corpus correspond an appropriate entry in the lexicon. In other words, lexical coverage is not evaluated here. We intend to evaluate hypotheses, not resources.

¹¹There were no examples of error expressions in the results concerning ambiguous *pp* forms.

The following is Smorph output for *Il la note*

```
'il' .
[ 'il' , 'TPRO' , 'pnom' , 'TPASS' , 'auf' ] .

'la' .
[ 'la' , 'TFG' , 'cl' , 'AMB' , 'ocl' , 'TPASS' , 'auf' ] .

'note' .
[ 'noter' , 'TFV' , 'v' , 'MOD' , 'fl' , 'AMB' , 'ov' ] .
```

where, besides the already commented values auf, fl, ov, the values pnom, cl, v, ocl stand for *nominative pronoun, clitic, verb, ambiguous clitic*, respectively.

5 Rules

Four basic types of rules are expressed in the same overall formalism:

1. Structural disambiguation rules of potentially ambiguous verbal forms (ov) which are actually not in chunks.
2. Rules for the construction of verbal chunks
 - (a) not potentially ambiguous,
 - (b) potentially ambiguous.
3. Rules for the disambiguation of potentially ambiguous chunks.
4. Rules for the disambiguation of forms not disambiguated by (1) to (3).

The rules are illustrated in the following in a semi-formal notation, where capital letters stand for variables and feature/value pairs are reduced to a minimum, *i.e.* a value. Table 1 gives the key to the symbols used in the examples.

Rules (1) concentrate on prepositional and nominal phrases, and on the initial sentence position.

As an example, the rule

$X[\text{det}] \ Y[\text{ov}] \ \rightarrow \ X[\text{det}] \ Y[\text{not-v}]$

specifies that an ambiguous verbal form following a determiner is not a verb (e.g. *ce juge*).

<i>Lexical forms</i>	
cl	clitic
ocl	ambiguous clitic
v	verb
ov	ambiguous verb
nv	not ambiguous verb
det	determiner
prep	preposition
pnom	nominative pronoun

<i>Chunks</i>	
vnfl	not ambiguous verbal inflected chunk
vnfla	ambiguous verbal inflected chunk

<i>Other symbols</i>	
not-	negation prefix
+	concatenation operator

Table 1: Key to symbols used in the example rules.

Rules (1) are not intended to be exhaustive, but have been reduced to what was considered to be a strict and sufficient minimum in order to obtain the proposed target. Thus, it is apparent that many potentially ambiguous verb forms (ov) remain: their potential ambiguity is not resolved.

Rules (2a) are specified around non ambiguous verbs (nv) and also around potentially ambiguous ones (ov), inasmuch these are contextually disambiguated internally in the verbal chunk by the presence of an unambiguous form (e.g. *ne*).

As an example, the rule

$$X[cl] \ Y[cl] \ Z[v] \ \rightarrow \ X+Y+Z[vnfl]$$

disambiguates *le*, *lui* and *donne* and constructs the chunk in *Jacques [le lui donne]_{vnfl}*.

Rules (2b) specify potentially ambiguous verbal chunks containing at least two forms. This situation typically arises when chunks are formed by a clitic pronoun and a verb, the two being ambiguous.

As an example, the rule

$$X[ocl] \ Y[ov] \ \rightarrow \ X+Y[vnfla]$$

specifies a provisional ambiguous chunk for *la note* or *en charge*.

A challenge for rules (2) is the identification of incidental phrases. Two basic types of incidental phrases were foreseen, one between commas and the other without, as in

- *Il a, trs souvent, t invit.*
- *Il a trs souvent t invit.*

They are respectively described by the two following structures making use of the Kleene + operator

, auf+ ,
auf+

These straightforwardly say : incidental phrases in verbal chunks are constructed by one or more auf expressions, which can be preceded and followed by commas. Depending on the verbal chunk, the point of insertion of the incidental phrase can be one or another. Following the general strategy, not all possible incidental phrases were defined in our rules, but the general hypothesis is that the same structures are candidates to express all of them.

Rules (2a) and (2b) are intended to be exhaustive within their respective scopes : Rules (2a) must build all non ambiguous verbal chunks and Rules (2b) all ambiguous verbal ones with two forms. After application of these rules, some verbal forms (the ov ones) and verb chunks (the ones built by Rules (2b)) remain thus to be disambiguated. This is the job of Rules (3) and Rules (4).

Rules (3) take advantage of the cyclic application of rules. After cycle I, the already specified vnfl can be used to disambiguate already specified vnfla and ov verb forms..

As examples, the rules

$$\begin{aligned} X[vnfl] \ Y[vnfla] &\rightarrow X[vnfl] \ Y[not-vnfl] \\ X[vnfl] \ Y[ov] &\rightarrow X[vnfl] \ Y[not-vnfl] \end{aligned}$$

will respectively be in charge of the resolution of *en charge* in *Il [n'est pas]_{vnfl} en charge d'une solution*, where *en charge* is not a verbal chunk, and of *compte* in *Il [tient]_{vnfl} compte d'une solution*, where *compte* is not a verbal form. By the same token but with opposite effects, the rules

$$\begin{aligned} X[pnom] \ Y[vnfla] &\rightarrow X[pnom] \ Y[vnfl] \\ X[pnom] \ Y[ov] &\rightarrow X[pnom] \ Y[vnfl] \end{aligned}$$

will respectively assign vnfl to *la juge* in *Il la juge* and to *compte* in *Il compte venir*.

It is at this point minimal statistics proceeds. Intuitively or using very simple heuristic techniques on very restricted texts, it is easy to know that, given the disambiguation performed by rules (1), ambiguous verbal chunks containing at least two forms ($vnfla$) are more frequent as nominal phrases than as verbal chunks (in particular as *le*, *la*, *les* are most likely to be determiners, and as *en* is more likely to be a preposition), and, conversely, that ambiguous verbal forms (ov) alone, excepted the jv ones, are more frequent as verbal chunks than as nouns. Rules (4) express this.

We operate thus with rules that can be called *structural*, *i.e.* Rules (1) to (3), and with remaining Rules (4), which are poorly heuristically motivated rules solving the “other” cases. Our system tags as $vnfl-I$ and $vninf-I$ inflected and infinitive verbal chunks obtained by structural rules, and as $vnfl-II$ and $vninf-II$ inflected and infinitive verbal chunks obtained by the remaining rules.

6 Results

Our general underlying hypothesis can be now precisely stated: limited lexical resources as characterized in Section 4, plus the linguistically motivated structural rules (Rules (1) to (3)), plus simple statistical/heuristic rules (Rules (4)) can reach results in the targetted level (*i.e.* around 98% recall and precision). Effectively obtained results, which, after the explicitation of the background for analysing them, will be presented below, do not falsify the hypothesis.

The obtained results must be analyzed in terms of the goals of the system on one hand, and in terms of the actual kinds of implemented and/or foreseen information on the other hand. We want to distinguish *missing resources*, which are foreseen but not implemented, from effective counter-examples to our hypothesis. E.g. if in a corpus there is an adverbial frozen expression or an incidental clause in a verbal chunk, foreseen but not implemented, we do not consider the errors resulting from such non-implementation as counter-examples to the hypothesis, but as missing resources.

We know, with respect to forms inside verbal chunks, that verbal forms and morphological items are exhaustively implemented, while adverbs and adverbial frozen expressions are not (see section 4). We know furthermore that Rules (1) are not intended to be exhaustive, while Rules (2a) and (2b) are intended to be so, even if not all formally foreseen incidental phrases are effectively implemented. We know finally that a subset of ambiguous verbal (ov) forms were classified on intuitive heuristic grounds as jv (see section 4), and that the $vnfla$ and ov forms, remaining as such after application of structural rules, are classified by Rules (4) as $non-v$ forms and $vnfl$ forms, respectively (see section 5).

From these, it is possible to classify counter-examples as *structural*, *i.e.* related to structural rules and *heuristic*, *i.e.* related to Rules (4). Structural counter-examples are further classified as being produced

- (i) by incompleteness of Rules (1),
- (ii) by minor inaccuracies in the formulation of Rules (2),
- (iii) by lack of the expressive power of Rules (2).

The limits of the system and its underlying general hypothesis are given by structural counter-examples produced by lack of expressive power of Rules (2) together with heuristic counter-examples. Intuitively, the system reaches its limits when it is not possible to modify it without modifying its general assumptions.

To evaluate the system, we will use the classic recall and precision measures. In addition to these measures, we define *SL* (System Limits) as :

$$SL = (1 - EE) * 100$$

where *EE* is the ratio of ‘error expressions’ in a corpus test over the number of chunks actually observed in the corpus (*i.e.* the denominator of the recall measure). An error expression is either an expression

		<i>possible</i>	<i>actual</i>	<i>correct</i>	<i>recall</i>	<i>precision</i>
training	infinitive	392	392	392	100.00	100.00
	finite	819	822	814	99.39	99.03
	all	1211	1214	1206	99.59	99.34
test state 1	infinitive	285	283	282	98.95	99.65
	finite	668	677	651	97.46	96.16
	all	953	960	933	97.90	97.18

Table 2: Results on training and test corpus in state 1.

		<i>possible</i>	<i>actual</i>	<i>correct</i>	<i>recall</i>	<i>precision</i>
test state 2	infinitive	285	284	284	99.65	100.00
	finite	668	673	661	98.95	98.22
	all	953	957	945	99.16	98.75

Table 3: Results on test corpus in state 2.

observed in the corpus test to which correspond one or more recall or precision errors¹² or an expression output by the system as a verbal chunk which do not correspond in part or in totality to an actually observed verbal chunk.

Given this background, it is now possible to report and analyse effective results of our system.

We used a 13,000 word corpus of finance news articles to tune our system during development. This corpus will be referred to as the “training” corpus. We emphasize that the “training” corpus is not a previously tagged one. It is used neither to extract statistical information on n-grams with $n \geq 2$ nor to learn rules from it. Structural rules and the lexical information which were needed were foreseen as working hypotheses. The contribution of the so-called “training” corpus is thus limited to the tuning of the system, to the verification of heuristic choices related to Rules (4) inasmuch they concern *ov* and *jv* forms, and to the detection of adverbs and adverbial frozen expressions incorporated to verbal chunks. See in Table 2 the results on the “training” corpus.

The system underlying the results of the “training” corpus was applied to a 10,400 word, previously unseen, “test” corpus of the same source. After this application, we detected 13 adverbial expressions which were missing in incidental clauses and added to the lexical resources. We refer to the state of the system at this stage as “state 1”. The system in state 1 was applied to both the “training” and the test corpora. Results are stated in Table 2. The *possible* and *actual* columns give the number of verbal chunks in the manually tagged corpus and in the output of the system, respectively.

Analysis of results on the test corpus reveals two missing (but formally foreseen) Rules (2) for inflected verbal chunks with incidental clauses, and four minor inaccuracies in the formulation of Rules (2) (two related to inflected chunks and two to infinitive ones). The two missing rules for inflected verbal chunks were added and the four verbal chunks rules modified. The system in State 2 was thus obtained. These modifications allowed the correct identification of 10 more finite verbal chunks and of 2 more infinitive verbal chunks (see Table 3). On the “training” corpus, the system in state 2 produces the same results as in state 1, except for one additional error.

¹²In the sentence, *Aucun groupe tranger n’a pour le moment fait connatre son intrt*, one observes the finite verbal chunk *n’a pour le moment fait*. Given this sentence, the system actually identifies two finite verbal chunks *n’a* and *fait*. We here have one recall error (the observed chunk is not identified) and two precision errors (the system identifies two chunks which are not observed as such in corpus), but only one error expression: *n’a pour le moment fait*. The recall and precision errors are correlated and this is taken into account in the computation of *SL*. Note that *SL* ranges from 100 (perfect result) to $-\infty$. Negative values are obtained when the system produces more error expressions than there are observed expressions in the corpus.

<i>label</i>	<i>correct</i>	<i>actual</i>	<i>precision</i>
vninf-I	273	273	100.00
vninf-II	11	11	100.00
vnfl-I	615	618	99.51
vnfl-II	46	55	83.64

Table 4: Precision figures for different rule types.

The analysis of error expressions in the results of the application of the system in state 2 to the test corpus allows to obtain a first estimation of *SL* and its underlying general hypothesis.

On the test corpus, 16 error expressions remain. They are associated to 20 recall or precision errors obtained by the system :

- 2 error expressions concern both recall and precision (i.e. 4 overall errors),
- 1 error expression ¹³ concerns 1 recall error and 2 precision errors (i.e. 3 overall errors),
- 8 error expressions each concern a precision error only,
- and 5 error expressions each concern a recall error only.

Out of 8 error expressions concerning precision errors, 3 arise from incompleteness of Rules 1. They affect finite verbal chunks but can be easily completed within the expressive power of the system. Thus they can be considered as counter-examples not affecting *SL*.

All of the other error expressions (i.e. 13) lead to mistaken results which are truly counter-examples counting for the calculation of *SL*.

The 3 error expressions affecting both precision and recall are produced by lack of expressivity of Rules (2), and as such, they are true counter-examples. They all involve finite verbal chunks with incidental clauses which are not marked by commas as in *L'encours ... [a pour sa part plus que doubl]*. The system cannot currently deal with such cases, because *part*, being a potential verbal form, is not an *auf* (see section 4). We know that these error expressions generate 7 recall or precision errors, all of them on finite verbal chunks.

The 10 remaining error expressions which are true counter-examples are all produced by statistical/heuristic failures. Only 1 of them is an infinitive verbal chunk (an error affecting recall). 4 affect the recall on finite verbal chunks and 5 the precision.

Thus we have the overall ratio of counter-example error expressions $EE \simeq 0.014\%$ (i.e. 13/953), which gives $SL \simeq 98.6\%$.

It is also interesting to remark that in the Smorph output of the test corpus there are 464 potentially ambiguous inflected verbal forms *ov* over 673 effective verb inflected chunks (69%), and 38 potentially ambiguous infinitive verbal forms *oin* over 284 effective verb infinitive chunks (13.4%). This is important because the challenge for disambiguation arises from *ov* and *oin* forms. In the test corpus, figures of *ov* tokens over *vnfl* forms (i.e. 464/673) practically inverse the ones in the lexicon of *nv* over *fl* forms (i.e. 429/659) (see section 4).

The error expressions which are true heuristic counter-examples can thus be reported to the challenge forms, i.e. 9/464 for *ov* forms, and 1/34, for *oin* forms, which gives $\simeq 1.9\%$ and $\simeq 2.6\%$ respectively. We claim thus that the encouraging *SL* obtained does not come from the fact that the challenging forms are weekly represented in the test corpus.

The verbal chunks obtained by the structural rules (1) to (3) are tagged *vnfl-I* and *vninf-I*: it means that these chunks are defined structurally as verbal chunks. The verbal chunks obtained by the

¹³The example in footnote 12 page 9.

rules (4) are tagged `vnfl-II` and `vninf-II`. These chunks are heuristically defined as verbal chunks. The glass box evaluation allows us to discriminate between the two sorts of tags and it is interesting to note that there is no significant difference between the precision obtained for both tags related to `vninf`. Table 4 reproduces the results obtained for these tags on the test corpus.

7 Perspectives

Neither all finite verbal chunks nor infinitive verbal chunks can be obtained by our system. However, it allows at this stage to reach a good level in the disambiguation task. Ongoing work deals with the evaluation of the cost, in terms of lexical resources and linguistic rules, to increase the results of the extraction of verbal chunks. It may be summarized by the following question: what kind and what quantity of new information does one need to significantly increase the results ?

Besides this, we also plan to:

- extend the system and develop our set of linguistic rules to perform the extraction of all verbal forms, including gerund and present participle chunks,
- further describe the potential incidental clauses within verbal chunks,
- extend the coverage of the system to verbal coordination of finite and infinitive verb, as well as participial forms in French,
- take advantage of collateral benefits of the system: if verbal chunks are correctly specified, we may incidentally deduce that all occurrences of ambiguous forms not in verbal chunks must not be associated to the function or the category to which they are associated in verbal chunks. Eg., all *lui* forms not in verbal chunks and not following a preposition must be associated to a nominative role ¹⁴.

Future work will also consist in testing the system on corpora from various sources and of different types, the goal being to isolate different sets of rules with respect to their precision and effectiveness in terms of recall. This next system could then become a modular system which the user might use with higher precision/lower recall or lower precision/higher recall, depending on his needs.

In other respects, we are also focused on the applicative dimension of a such system, and one application domain in which one would need high precision is teaching. In the field of language acquisition, our system may be used to set up a large variety of exercises using local contexts. The rule-based system, provided it is reliable, and the evaluation system can also be used in order to automatically detect the errors made by the students in the exercises. By integrating this system in a specific software, we are searching two goals: one is to link development in Natural Language Processing with the issue of French grammar teaching to foreign students and the other is to share the different concerns of NLP and teaching methods.

References

- [1] Abney, S. Part-of-Speech Tagging and Partial Parsing. In Church, K., Young S., Bloothoof G. (eds.): *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht (1996)
- [2] At-Mokhtar, S. *L'analyse prsyntaxique en une seule tape*. Ph.D. thesis Universit Blaise-Pascal/GRIL, Clermont-Fd (1998).

¹⁴In Bès & Dahl [3], verbal chunks along with phrase introducers (e.g. complementizers, interrogative expressions or conjunctions) are crucial for the obtention of coordination of verbal phrases.

- [3] Bs G.G., Dahl V. Balanced parentheses in NL texts: a useful cue in the syntax/semantics interface. In *Proceedings of the Lorraine-Saarland Workshop Series*, Nancy (2003).
- [4] Brill, E. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. In *Computational Linguistics*, 21 (1995), 543–565.
- [5] Chanod, J.-P., Tapanainen P. Tagging French - comparing a statistical and a constraint-based method In *Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*, Association for Computational Linguistics, Dublin, 1995, 149–156.
- [6] Charniak E. Statistical Techniques for Natural Language Parsing. In *AI Magazine*, 18(4), 1997, 33–44.
- [7] Habert B., Nazarenko A., Salem A. *Les linguistiques de corpus*. Paris, Armand Colin, 1997.
- [8] Manning Ch. D., Schtze H. *Foundations of statistical natural language processing*. Cambridge Mass, The MIT Press, 1999.
- [9] Paulo J.L., Mamede N.J., Hagge C. *PAsMO – Pos-AnaliSe MOrfologica*. Technical report, L2F - INESC-ID, Lisboa (2001).
- [10] Tapanainen P., Jrvinen T. Syntactic analysis of natural language using linguistic rules and corpus-based patterns. In *15th International Conference on Computational Linguistics Proceedings*, Kyoto, 1994, 629–634.
- [11] Tapanainen P., Voutilainen A. Tagging accurately – Don't guess if you know. In *Proceedings of the Fourth ACL Conference on Applied Natural Language Processing*. Association for Computational Linguistics, Stuttgart (1994) 47–52.
- [12] Trouilleux F. *Eva Pasmo – Un ensemble de scripts pour l'valuation de PASMO*. Technical report, GRIL – Universit Blaise-Pascal, Clermont II (2004)
- [13] Voutilainen A. EngCG tagger, Version 2, In Brondsted T., Lytje I. (eds.). *Sprog og Multimedier*. Aalborg Universitetsforlag, Aalborg (1997).