Non Deterministic Chunking

François Trouilleux

Non Deterministic Chunking

François Trouilleux
Clermont Université, Université Blaise Pascal,
EA 999, Laboratoire de recherche sur le langage,
BP 10448, F-63000 Clermont-Ferrand
francois.trouilleux@univ-bpclermont.fr

Abstract

This paper presents a non deterministic chunker for French. It is implemented in NooJ and operates on untagged text. The grammar is designed so as to consist only in a description of chunk composition; no other contextual information is used to disambiguate the chunks. The problem of the massive over-generation of chunks in an *All matches* pattern matching mode is dealt with using the NooJ +UNAMB feature. This feature is systematically used on left-hand side function words and as such expresses a fundamental property of French chunks. The resulting chunker obtains almost perfect recall and 73.52% precision on the development corpus and provides a tool to explore different levels of ambiguity.

While rule-based chunkers are generally conceived as defining patterns which are applied deterministically on part-of-speech tagged text, the present work explores the feasibility of a non deterministic chunker, which would operate on untagged text. The case study is on chunking French text.

Section 1 recalls the definition of chunks and introduces a number of observations about the nature of chunks in French. We discuss different approaches to rule-based parsing, specifically chunking, in section 2, and expose the motivations for our work. Section 3 introduces our data set and shows the major problems one faces when trying to match an ambiguous text to patterns. We propose a solution to these problems in section 4, in the form of a new grammar application mode. A good approximation of this grammar application mode is possible in NooJ thanks to the +UNAMB feature. This feature is used in our grammar to favour the interpretation of ambiguous left-hand side function words as such, thus taking advantage of the very nature of French chunks.

1 On chunks in general and in French in particular

1.1 Definitions

Chunks were first defined by Abney (1991) in the following terms:

> I define chunks in terms of major heads. Major heads are all content words except those that appear between a function word *f* and the content word that *f* selects. For example, *proud* is a major head in *a man proud of his son*, but *proud* is not a major head in *the proud man*, because it appears between the function word *the* and the content word *man* selected by *the*.

Hence, there are three chunks in *[a man] [proud] [of his son]*, but only one in *[the proud man]*.

In this definition, the key concepts are that of function word and content word. Distinguishing the two is not a problem in the general case, but there are two notably unclear cases:

- pronouns: Abney considers them as function words, but view them as major heads if they are selected by a preposition
- auxiliary verbs: Abney's examples show that he considers them as function words; they are not in the EASY annotation scheme, where *[il est] [mangé]* ("he is eaten") makes two chunks.

The annotation scheme used in the EASY evaluation campaign of French parsers (Gendner and Vilnat, 2004) provides six categories for chunks: GN (noun group), GP (prepositional group), GA (adjectival group), GR (adverbial group), PV (verb group with a preposition) and

NV (other verb groups). We use this annotation scheme as a reference, except that in the work presented here the adjective-past participle ambiguity is not dealt with. Past participles are thus systematically analysed as GAs, rather than NVs. This is motivated by the fact that we work only on the composition of chunks and, as GAs and NVs with past participles are always unary in the EASY scheme, there is no way to disambiguate between the two.

Figure 1 provides an example of an EASY style chunked text.[1]

<GP>Pour cent francs</GP> <GP>par an</GP>,
<NV>elle faisait</NV> <GN>la cuisine</GN>
et <GN>le ménage</GN>, <NV>cousait</NV>,
<NV>lavait</NV>, <NV>repassait</NV>,
<NV>savait</NV> <NV>brider</NV> <GN>un
cheval</GN>, <NV>engraisser</NV> <GN>les
volailles</GN>, <NV>battre</NV> <GN>le
beurre</GN>, et <NV>resta</NV>
<GA>fidèle</GA> <GP>à sa maîtresse</GP>...

Figure 1. An EASY-style chunked text.

1.2 Observations

Given Abney's definition and the EASY evaluation scheme, we report here a number of observations which are worth remembering to support our project.

First, we note that the EASY annotation scheme is deliberately designed so as to never yield an embedding of one chunk into another.[2] A chunked text is a flat sequence of chunks, conjunctions and punctuation signs. As is well know, these chunks may be described by regular grammars.

Second, we observe that chunking is akin to morphology, as the examples in Table 1 show.

isolated words		*affix-like spelling*
give **it** to **me**	donne-**le**-**moi**	dá**melo** (es)
a sea **the** sea **the** blue sea	une mer **la** mer **la** mer bleue	mare (ro) mare**a** albastr**a** mare
in the house	**dans** la maison	a ház**ban** (hu)

Table 1. Different ways to spell the same information.

As far as writing is concerned, what is expressed with isolated words in English (column 1) or French (column 2) is expressed with affixes in other languages (column 3), e.g. pronouns in Spanish, definiteness in Romanian, or the equivalent to the preposition *in* in Hungarian. To some extent, chunking may be viewed as the task of combining function words to content words in the same manner as morphology is combining affixes to word bases, *i.e.* agglutinating function words in non-agglutinative languages.[3]

Finally, we observe that in French the vast majority of chunks can be described with the pattern FW* CW, *i.e.* a sequence of 0 to *n* function words followed by one content word. In principle, function words to the right of the content word in French should be written down with a hyphen, as in *donne-le-moi* ("give it to me"). This, in a sense, confirms the fundamental FW* CW pattern. Note that all the chunks in Figure 1 are described by this pattern (function words are underlined).

In themselves these observations are not new, but they will support the approach to chunking we will develop in sections 4 and 5.

2 Approaches to chunking

Parsers may be classified according to their goal, shallow *vs*. deep analysis, and the method to reach it, rule-based *vs*. statistical. We focus here on rule-based parsing. In this category, one can make a distinction between parsers, depending on whether the rules are applied on part-of-speech disambiguated text or not. Combined with the shallow/deep analysis distinction, this defines four options.

The first option is to build a deep parser working directly on ambiguous text. This is the most classical goal in syntax. Examples of such parsers for French are in (Boullier *et al*., 2005) and (Goldman *et al*., 2005). Deep parsers, as is well known, are confronted with massive ambiguities. Even though recent advances have been made in this matter (see Boullier *et al.*, 2005), this may lead to computation problems. As a result (second option), parser developers often use part-of-speech taggers to first disambiguate the text, thus reducing the complexity of the analysis process. As an example, Roussanaly *et al*., (2005) motivate the use of a tagger in the analysis chain they

[1] "For a hundred francs a year, she cooked and did the housework, sewed, washed, ironed, knew how to harness a horse, fatten the poultry, make the butter, and remained faithful to her mistress..."

[2] See section B.3 in (Gendner and Vilnat, 2004).

[3] Considering two different types of function words, clitics and non-clitics, could possibly be useful. We do not know of any approach to chunking that uses this distinction, however.

developed for the EASY campaign by the "crippling processing time" caused by "multiple ambiguities". This text tagging strategy was also adopted by several other participants to the EASY campaign, in particular, among the parsers described in (Jardino, 2005), Syntex, TagParser, LIMA and the LPL parsers.

If one considers shallow parsers, by far the most common approach is also to process part-of-speech disambiguated text. There are many examples of this approach: (Hindle, 1994), (Kinyon, 2001), (Aït-Mokhtar *et al*., 2002), (Bourigault *et al*., 2005), etc. The following definition of a chunker, taken from the Natural Language Toolkit documentation (Bird *et al*., 2009, section 7.2), is illustrative of the paradigmatic nature of this approach:

> A chunker finds contiguous, non-overlapping spans of related tokens and groups them together into chunks. Chunkers often operate on tagged texts, and use the tags to make chunking decisions.[4]

We here want to follow the fourth option, that of a shallow parser which operates on ambiguous text.

With respect to the deep analysis goal, we simply want to decompose the ambiguity problem, and focus on a simpler one.

With respect to first tagging text, we want to avoid that because (1) the determinism of the process is inadequate, as real ambiguities exist and the information to deal with some spurious ambiguities is not always available, and (2) the linguistic description in a two step incremental process is redundant: for instance the tagger will record that a verb reading is more likely than a noun reading after a subject pronoun, but this information will in some way also be included in the verb chunk description.

The goal of this work is thus to experiment with chunking ambiguous French text. NooJ is perfectly adequate for this task, as it provides a general, theory neutral framework for morphological and syntactic analysis.

Recently, Vučković *et al*. (2008) proposed a NooJ chunker for Croatian, while Fay-Varnier *et al*. (2008) used NooJ to chunk French texts. A non deterministic chunker for French is also presented in (Trouilleux, 2009). The approach experimented here differs from those in that chunks will be identified by a single grammar – as opposed to the rule cascade of (Vučković *et al*., 2008), and with patterns which do not include any contextual information – as opposed to the patterns used in (Trouilleux, 2009). This will result in a simpler grammar which will better account for the specific nature of chunks.

3 Chunking an ambiguous input

The characteristics of French chunks we observed in section 1.2 lead us to propose a basic grammar model for their identification in a corpus. We here present our data set, the results obtained by this basic model, and an inventory of the problems encountered.

3.1 Data and results

To perform the analysis presented below, we used a grammar which specifies the six types of chunks, as the one in Figure 2. Each of the non terminals GP, GN, etc. specifies exactly one type of chunk and annotate it. No context is considered.[5]

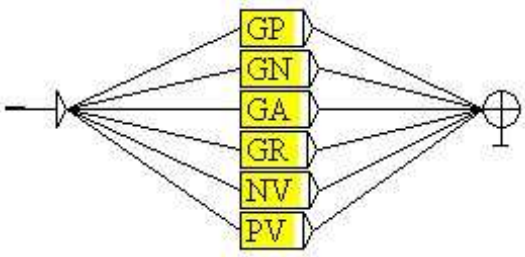

Figure 2. The main grammar graph

Let us assume this grammar is correct with respect to chunk composition. Actually, if one considers the specific 22,557 word corpus used to develop this grammar, it does identify all but one (an English title) of the 11,144 chunks. This corpus is composed of Flaubert's *Un cœur simple* (11,581 words, 51%), 157 examples taken from the *de* entry of the TLF (5,309 words, 23.5%), the transcription of the Jean-Claude Mery tape (3,449 words, 15.5%), and nine articles from *La Tribune* (2,218 words, 10%).

NooJ offers three ways to apply a grammar, depending on whether one wants to select the longest matches, the shortest matches or all matches. Table 2 gives the recall and precision measures obtained by our grammar on our corpus using each of the three application modes.

[4] The NLTK chunker indeed operates on tagged texts.

[5] Besides the obvious ADJ, ADV, N, V, DET, PREP lexical categories, we use AUX for auxiliary verbs, CL for clitic pronouns, PRO for other pronouns, NUM for cardinal numerals, NEG for the negation particle *ne*, and NEGPAS for the negation adverb *pas* and the like.

	Shortest	Longest	All
recall	70.22	97.90	99.99
precision	38.78	77.30	33.06
f-measure	45.77	86.39	49.72

Table 2. Results obtained with different grammar application modes

Recall is close to 100% in the All matches mode but precision is very low: 33.06%. This is annoying because, obviously, if one wants to preserve some ambiguities in the output of the chunker, the deterministic Shortest and Longest matches modes are inadequate. The results observed for these two modes, however, show that the Shortest matches mode is clearly inadequate, whereas the Longest matches mode yields quite good recall, if not perfect, and much better precision as the All matches mode. The Longest matches mode is clearly the best on F-measure.

3.2 All matches problems

Compared to the Longest matches mode, there are two main problems with the All matches mode. One comes from the fact that, as grammars are applied in a pattern matching fashion, the system may leave some lexical units out. Hence, for instance, for the sequence *il la porte* ("he carries it"), as shown in Figure 3, the system not only correctly identifies the whole sequence as a NV (line 1), but also provides five other analyses which leave out *il* or *la*. While recall is 1/1 = 100%, precision is 1/6 = 16.7%.

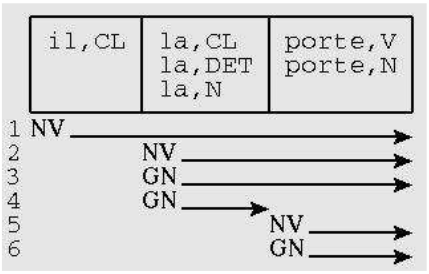

Figure 3. All matches for *il la porte*.

This problem can be characterized as one of coverage: as chunks cover the whole text except for conjunctions and punctuation signs, no lexical unit other than those should be left out.

The other major problem, still compared to the Longest matches mode, comes from some very frequent ambiguities involving forms which are ambiguous between function and content words. Table 3 shows 8 examples of such ambiguities, taken from the 15 most frequent lexical ambiguities recorded by NooJ in our corpus.[6]

freq.	*form*	*categories*
683	la	DET CL N
275	une	DET N
209	dans	PREP N
156	pour	PREP N
133	pas	NEGPAS N
133	son	DET N
124	est	AUX N
117	par	PREP N

Table 3. Most frequent functional/content word ambiguities.

All these forms can be nouns, a noun alone can form a GN, so each occurrence of these forms yields a spurious GN. However, the functional reading is by far the most frequent, since the corpus contains only 12 noun readings for these forms: 10 *pas* and 2 *est*.

3.3 Longest matches problems

Compared to the All matches mode, the Longest matches mode yields 233 extra recall errors. There are three types of errors. Examples are given in Table 4.

The most frequent error type (line 1, 203 errors, 87%) involves an extension of the right frontier of GNs and GPs one content word too far.

The second error type (line 2, 28 errors, 12%) has to do with the word forms *tout* and *toute*, which are ambiguous between adverb or determiner, or, for *tout*, pronoun. In the EASY annotation scheme, when they are determiners, *tout* and *toute* are included in a GN or GP, while when they are adverb or pronoun they are kept out of the GA or NV chunk. Thus while *tout change* and *toute surprise* should be analysed as two chunks, the Longest matches mode favours a GN reading.

Finally, 2 recall errors (line 3 of Table 4) are due to a punctuation problem: in spoken form, the sentence *En voilà une Mme Lehoussais, qui au lieu de prendre un jeune homme…*[7] would necessarily be uttered with an intonation which

[6] See footnote 5 for information on the categories. We include the auxiliary verb *est* ("is") in these examples, even though it is not a function word in the EASY annotation scheme. Other ambiguities in the 15 most frequent involve two function word readings, e.g. *le* as a determiner or a clitic pronoun, *en* as a preposition or a clitic pronoun, etc.

[7] "Here's one, Mrs Lehoussais, who instead of taking a young man…"

	expected analysis	*system output*
1	<GN>des bas</GN> <GA>gris</GA> en. *gray stockings* <GN>le surnaturel</GN> <NV>est</NV> tout simple en. *the supernatural is very simple*	<GN>des bas gris</GN> en. *low grays* <GN>le surnaturel est</GN> tout simple en. *the supernatural east…*
2	<GN>tout</GN> <NV>change</NV> en. *everything changes* <GR>toute</GR> <GA>surprise</GA> en. *most surprised*	<GN>tout change</GN> en. *any exchange* <GN>toute surprise</GN> en. *any surprise*
3	En voilà <GN>une</GN> <GN>Mme Lehoussais</GN> en. *Here's one Mrs Lehoussais*	En voilà <GN>une Mme Lehoussais</GN>

Table 4. Longest matches mode recall errors.

would clearly separate *une* and *Mme Lehoussais*; Flaubert did not mark this specific intonation, but a comma would be quite appropriate here and we consider this error is beyond the scope of the present work.

The first two error types are different in that the first involves a sequence of two content words, while the second involves forms which are ambiguous between functional (*tout* as a determiner) and content words (*tout* as adverb or pronoun). In fact, the second type of error could be avoided in the Longest matches mode simply by always viewing *tout/toute* as functional, and hence have the following reference annotation:

```
<NV>tout change</NV>
<GA>toute surprise</GA>
```

In addition to the algorithmic considerations developed here, we see two arguments in favour of this analysis: one is the fact that it would be in line with Abney's analysis for the pronoun reading (see section 1.1), the other is the fact that orally, in both cases, a liaison is required after *tout* [tu] if the verb or adjective starts with a vowel, e.g. *tout habillé* [tutabije].

Adopting this point of view, we can make the following observations:

- the Longest matches mode is adequate to account for the inclusion of left-hand side function words into chunks,
- the All matches mode is required to account for possible ambiguities with two content words to the right of GNs and GPs.

We then suggest that a new, intermediate, grammar application mode would be appropriate.

4 Towards a new grammar application mode

In order to allow complete identification of chunks (100% recall) with reasonable precision, using only a description of the chunks composition, we suggest that a new grammar application mode could be used. It relies on the distinction between function and content words, which, as we have seen, is fundamental to chunk definition (see section 1.1), and on the fact that left-hand side function words should be read using the longest matches mode.

Given a grammar, assume one has the set of declaratively specified left-hand side function words (LF). For instance, for French: [8]

```
<PREP> + <CL> + <DET> +
<NUM> + <NEG>
```

Let an *initial* LF be a left-hand side function word which comes before the head of a chunk. Given these two definitions, our chunking mode is defined as follows:

> For each text unit, select the sets of segments which maximize both the number of initial LFs and input coverage.

Coverage is defined as the number of lexical units which are included in a chunk.

For the *il la porte* example in Figure 3, it is easy to see that the correct analysis scores 3 in coverage and 2 in the number of initial LFs (ILF), while all other analyses, which leave *il* out, will score at most 2 coverage and 1 ILF.

[8] Prepositions, clitic pronouns, determiners, cardinal numerals, and the negation particle *ne*.

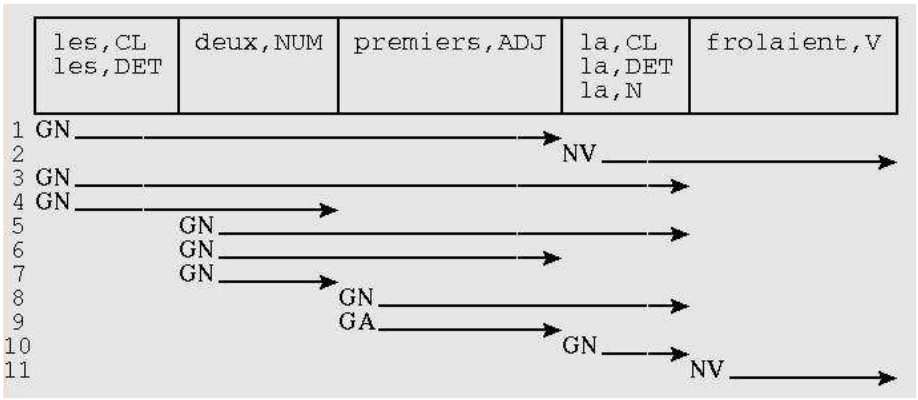

Figure 4. All matches for *les deux premiers la frôlaient*.

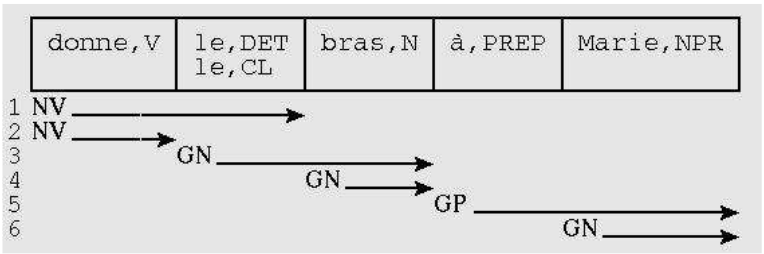

Figure 5. All matches for *donne le bras à Marie*.

Figure 4 gives all the matches obtained for the sequence *les deux premiers la frôlaient* ("the fist two were brushing against her"). The first two arrows correspond to the correct analysis and are indeed selected by our definition. Arrows 5, 6 and 7 are discarded as they imply leaving *les* out. Arrow 10 is discarded as it can only be used with 11 and 2 yields better #ILF than the 10-11 combination (include *la* in NV). Arrow 1 is better than the 4-9 combination, as it score 2 ILF (*les* and *deux*), while 4-9 scores 1+0.[9] Similarly, 3 is better than 4-8. Finally, two combinations remain, 1-2 and 3-11; 1-2 is preferred because it contains three ILF (*les*, *deux*, *la*) while 3-11 contains only two.

In Figure 4, the functional reading of *la* is preferred to the lexical one; Figure 5 gives another example where the alternative is between two functional readings, left-hand *vs*. right-hand side. We noted (section 1.2) that in principle right-hand side function words in French should be attached to the preceding word by a hyphen. However, robust corpus analysis should account for possible omission of the hyphen. Hence, *donne le* (imperative "give it") should be analysed as a possible NV. In the analysis for the sentence *donne le bras à Marie* ("give the arm to Mary"), there are two possibilities to maximize coverage: 1-4-5 and 2-3-5. The 2-3-5 (correct) combination is preferred because 3 contains 1 ILF while 1 and 4 do not. This accounts for the fact that the function word *le* appears preferably on the left-hand side of a chunk.

Note that the sentence *donne le à Marie* ("give it to Mary") would also be correctly analysed, as maximizing coverage would imply including *le* as a clitic in *donne le*.

5 Using the NooJ +UNAMB feature

Following the proposal of this new grammar application mode at the NooJ 2009 conference in Tozeur, Max Silberztein suggested using the NooJ +UNAMB feature instead.

5.1 Definition

Use of the +UNAMB feature in grammars is illustrated in section 15.8 "Special feature +UNAMB" of the manual, with an example where it appears only on the last state of a graph. More generally, its interpretation can be characterized as follows:

> Starting at a given lexical unit, if there is one or several path(s) with one or several +UNAMB feature(s), select the path(s) with the highest number of +UNAMB features.

[9] The cardinal numeral *deux* is not an initial LF in *les deux* as it is the head of the chunk.

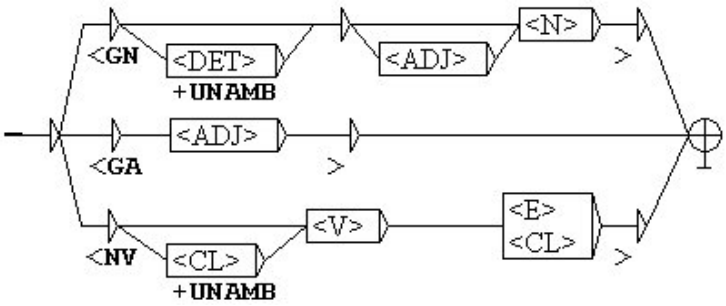

Figure 6. A sample grammar identifying three very basic chunks.

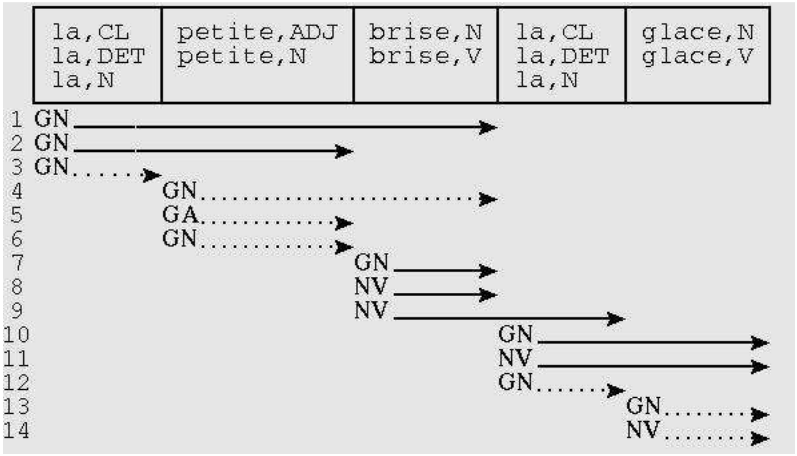

Figure 7. Selected and filtered out matches for *la petite brise la glace*.

> Move on to the lexical unit which follows the shortest of the selected paths and start again.

In this definition, the idea of selecting the paths with the highest number of +UNAMB features is a new NooJ feature proposed by Max Silberztein at the NooJ 2009 Conference, and implemented afterwards.

As an illustration of this mechanism, consider the grammar in Figure 6. It contains two +UNAMB features, which are set on the paths through initial left-hand side function words of category DET and CL. Applied to the sentence *la petite brise la glace*,[10] this grammar will produce the chunks marked by solid arrows in Figure 7, while the chunks marked by dotted arrows are those which would have been also identified if the +UNAMB features had not been used. Starting from *la* the system finds 1, 2, and 3, discard 3, moves on to *brise* (end of the shortest path selected), finds 7, 8 and 9, moves on to *la*, finds 10, 11 and 12, discard 12, and reaches the end of the analysed string.

Compared with the proposition we made in section 4, the only difference is that the +UNAMB method identifies a NV chunk over *brise la*.[11]

5.2 Application

To implement an approximation of the grammar application mode proposed in section 4, we added +UNAMB features on initial function words in the sub-graphs of our Figure 2 grammar. As an example, Figure 8 shows the subgraph which identifies PVs and infinitive NVs.[12] Using this grammar to parse the text in All matches mode, we obtain the results in Table 5 (*after* column).

It must be noted that, in addition to the categories listed as initial function words in section 4, we also had to mark as +UNAMB the adjectives

[10] This classic French example has two interpretations corresponding to the combinations 1-11 ("the little breeze freezes her") and 2-8-10 ("the little one breaks the ice") in Figure 7.

[11] See the discussion on Figure 5 in the previous section, for an example of why and how this chunk would be discarded.

[12] A direct comparison of this graph to the one in (Trouilleux, 2009) can be made: +UNAMB features have been added, and right-hand side nodes which allowed the identification of *sequences* of chunks have been removed.

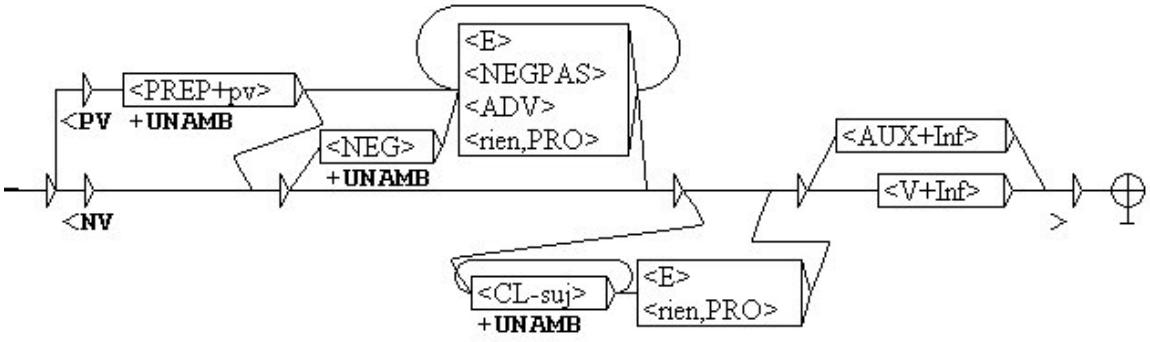

Figure 8. The PV sub-graph, to identify PVs and infinitive NVs.

tout and *tel* within GN and GP, a set of adverbs coming before adjectives (e.g. *très*, *plus*, and a few adverbs which combine with preposition de (e.g. *près*, *autour*). The grammar also contains +UNAMB features to handle long proper names and the deterministic attachment of right-hand side clitics with hyphens.

	before	*after*
recall	99.99	99.83
precision	33.06	73.56
f-measure	49.72	84.70

Table 5. Results using +UNAMB

Compared to the results initially obtained with the All matches mode (*before* column), there are 18 additional recall errors. 2 are due to the *Mme Lehoussais* sequence (see Table 4, line 3), the other 16 are due to complex adverbs such as *en particulier, à peine, en fait.*[13] These are declared in the dictionary as multiword units and at the grammar level they are read either as <ADV> (*i.e.* content words), or as a sequence function word + content word.[14] The latter interpretation is selected, as the function word is marked +UNAMB.

A possible way of dealing with such cases would be to mark multiword units starting with a function word with a special feature in the dictionary so that they can be distinguished in the grammar with a +UNAMB feature. Solving this problem would raise recall to 99.97%, with 73.52% precision. In terms of F-measure, we are getting close to the Longest matches results (see Table 2), with clearer possibilities to further improve the new results.

Compared to the results obtained by our grammar with patterns annotating sequences of chunks (Trouilleux, 2009), precision is approximately 2 points lower. Besides the fact that a few compounds are now segmented ambiguously, this is due to the fact that the extended patterns of (Trouilleux, 2009) not only handle the inclusion of function words into chunks, but also express preferences for some chunk sequences, in particular in favour of a GA (adjective or past participle) after an auxiliary verb. Even though precision is currently slightly lower, we consider this new grammar is better than that of (Trouilleux, 2009). Its design is much simpler and integration of the descriptions it contains in a larger description at the clause level will be easier.

6 Conclusion and perspectives

There are two major ways to obtain chunks in a rule-based approach: either one considers the whole sentence, going for a full parse, and chunks are a result of the global analysis, or one first tags the text and builds chunks deterministically using pattern matching techniques. The chunker presented here shares untagged input with the first approach and pattern matching with the second. Pattern matching techniques offer robust and fast corpus analysis, untagged text is closer to real life situation.

The chunker is built around a principle: the grammar only consists in a description of the chunks. Thus the results we obtain with it show to what extent chunk composition may disambiguate the text, and where the remaining ambiguities lie. As such, we view this chunker as a tool to explore the way different types of ambiguities are resolved. In this respect, the next steps will be to experiment with the adjective/noun ambiguity, the GN-GP ambiguity which results from the ambiguity of *de*, *du*, *de la*, *des*, and the NV-GN ambiguity.

The results presented here are on a specific, relatively small corpus which has been used

[13] "in particular", "hardly", "in fact".
[14] e.g. <PREP> <N>, or <CL> <V> for *en fait*.

throughout the development of the grammar. Larger scale evaluation of this chunker will of course be required. We see two directions in this respect: one is the classical global evaluation on an unseen corpus, the other is to test the approach on a carefully designed test suite. We will rather focus on the second type of evaluation. In particular, one goal will be to collect a set of examples involving forms used as content words while they may also have a functional reading (e.g. *la* meaning the A note, *son* meaning "sound", *vers* meaning "verse" or "worms", etc.). As we have seen in section 3.2, there are very few occurrences of such readings in our corpus, and while we can be quite confident that our grammar will identify functional uses of these forms, we still have to show that it will perform as well on content word uses.

Acknowledgments
Many thanks to Max Silberztein for his work on NooJ.

References

Steven Abney. 1991. Parsing by chunks. In R. Berwick, S. Abney & C. Tenny, Eds., *Principle-Based Parsing*. Kluwer Academic Publishers, Dordrecht.

Salah Aït-Mokhtar, Jean-Pierre Chanod and Claude Roux. 2002. Robustness beyond shallowness: incremental dependency parsing. *Natural Language Engineering*, 8, 121–144.

Steven Bird, Ewan Klein and Edward Loper. 2009. *Analyzing Text with Python and the Natural Language Toolkit*. http://www.nltk.org/book.

Pierre Boullier, Lionel Clément, Benoît Sagot, Eric Villemonte de La Clergerie. 2005. Simple comme EASy. In (Jardino, 2005), p. 35–39.

Didier Bourigault, Cécile Fabre, Cécile Frérot, Marie-Paul Jacques and Sylwia Ozdowska. 2005. Syntex, analyseur syntaxique de corpus. In (Jardino, 2005).

Christine Fay-Varnier, Qiuyue Li, Azim Roussanaly. (2008). Using NooJ's to parse constituents in the French PASSAGE corpus. Communication at the NooJ 2008 Conference. http://www.nytud.hu/nooj08/programme.html

Véronique Gendner and Anne Vilnat. 2004. *Les annotations syntaxiques de référence PEAS.* http://www.limsi.fr/Recherche/CORVAL/easy/.

Jean-Philippe Goldman, Christopher Laenzlinger, Gabriela Soare and Eric Wehrli. 2005. L'analyseur syntaxique multilingue FiPS dans la campagne EASy. In (Jardino, 2005), p. 35–39.

Donald Hindle. 1994. A parser for text corpora. In B.T.S. Atkins and A. Zampolli, Eds., *Computational Approaches to the Lexicon*, p. 103–151. Clarendon Press, Oxford.

Michèle Jardino, Ed. 2005. Actes de TALN 2005 (Traitement automatique des langues naturelles), Dourdan. ATALA, LIMSI.

Alexandra Kinyon. 2001. A language-independent shallow-parser compiler. In ACL, p. 322–329.

Azim Roussanaly, Benoit Crabbé and Jérôme Perrin. 2005. Premier bilan de la participation du LORIA à la campagne d'évaluation EASy. In (Jardino, 2005), p. 49–52.

Max Silberztein. 2004. Nooj: an oriented object approach. In J. Royauté and M. Silberztein, Eds., *INTEX pour la Linguistique et le Traitement Automatique des Langues*. Presses Universitaires de Franche-Comté.

François Trouilleux. 2009. Un analyseur de surface non déterministe pour le français. *Actes de TALN 2009*, Senlis, 24–26 juin 2009.

Kristina Vučković, Marko Tadić and Zdravko Dovedan. 2008. Rule-Based Chunker for Croatian. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco