*Revue, Informatique et Statistique dans les Sciences humaines*
XXII, 1-4, 1986, pp. 7-22.

# Using a Latin computer corpus for linguistic research

J.R. DE JONG and P.C. MASEREEUW

## 0. Introduction

Necessity (the lack of native speakers) has caused Latin linguistics to be a largely corpus oriented branch of linguistics. Many means have been developed to provide the linguist with access to the corpus of Latin texts, among which lexica, grammars, the Thesaurus Linguae Latinae etc. Recently, it has become possible to add the computer to this list of tools. In this paper we shall demonstrate how the computer can be put to use for research in the field of Latin linguistics.

In section 1, we shall describe how a (LASLA based) computer-corpus was obtained and implemented at our university. In section 2, we shall give an outline of the software available for access to computer corpora. In section 3, we shall describe our efforts to enhance the usefulness of the corpus.

## 1. Implementation

Computer corpora are essentially of two kinds. They either contain 'bare' texts, such as they are found in their written source, or the text plus additional information concerning, for example, the grammatical analysis of each of the words in the corpus. The former are more readily available for a number of languages, the latter, as a consequence of the much greater human effort required for their production, are fewer in

number. For English, the Brown and LOB corpora are examples of the latter kind. Each word in these corpora is followed by a code containing morphological information. To our knowledge the corpus produced at the *Laboratoire d'Analyse Statistique des Langues Anciennes* at Liege is the only sizeable Latin corpus of this type. When we considered the purchase of a Latin corpus, the promises that the additional coding seemed to hold for linguistic research outweighed the considerable extra cost per word that was involved. In 1980, our department acquired a subset of the LASLA corpus. In view of the type of research carried out at our department we confined our corpus mainly to prose authors. The texts presently at our disposal are[1]:
Caesar, *De Bello Gallico*;
Cicero, *Catilinariae, Pro Lege Manilia, Pro Caecina, De Natura Deorum* 3, *Tusculanae Disputationes*;
Curtius Rufus, 3 - 8,6,1;
Livy, extracts, ca. 50 chapters;
Plautus, *Amphitruo, Aulularia, Bacchides, Captivi, Casina*;
Sallust, *De Catilinae Coniuratione, Bellum Iugurtinum*;
Tacitus, *Agricola, Dialogus de Oratoribus*;
Virgil, *Aeneid* 1-6, *Georgics, Eclogues*.
The total number of words in the subset is about 300,000. We shall henceforth refer to this subset as the 'LASLA corpus'.

The LASLA corpus contains three kinds of information :
- The TEXT-FORMs : the words such as they appear in the printed text;
- The LEMMAta: with each text-form, a lemma is specified from which the text-form was derived, with numerical tags distinguishing homonyms;
- The ANALYSIS: a code specifying the morphological analysis of the text-form : its lexical category and in addition, with nouns declension case and number, with verbs conjugation, person, number, mood, tense etc. Similar, but less elaborate distinctions are made for other categories.

The corpus contains hardly any syntactic information : no information regarding the syntactic function of constituents (Subject, Object, Attribute, etc.), nor any indication for

1. A few texts are not yet in use, due to deviating or unfinished coding or technical problems : Cicero, S. Rab. Perd.; Vitruvibus, 2 bks; Plautus, Asin.

syntactic boundaries, such as clausal or phrasal boundaries (except the end of the sentence). One type of syntactic information which has been included concerns subordinate clauses : to each verb serving as a Predicate in a subordinate clause a code has been assigned specifying the type of subordination : infinitival, participal or finite, the last with different codes for different governing conjuctions.

The fact that the corpus was mainly designed for lexicographic and statistic purposes has left its mark: the corpus contains no interpunction, except that words followed by a period or question mark in the printed text are marked as sentence-final. There is no trace of commas, colons or semi-colons (the latter two are not interpreted as sentence-demarcators), which to some extent impairs the readability of output text.

Another feature of the corpus was that it turned out to be impossible to reconstruct in a direct way the original text from the text-forms of the words such as they appear in the corpus. Putting text-forms in a row, the initial words of Caesar's *Bellum Gallicum* would read as follows : *Gallia est divisa omnis est divisa* ... The second word in the text (*est*) was coded as derived from the lemma SVM, appearing in the text-form *est divisa*. The fourth word belongs to the lemma DIVIDO, but its text-form was again *est divisa*. A complex expression such as *non solum ... sed etiam* was treated in a similar way : the lemmata are NON, SOLVM, SED and ETIAM, respectively, but the text-form was *non solum sed etiam,* repeated four times. For the lexicographer, whose attention is directed at words rather than sentences, this may be a logical way to proceed, but for other linguistic purposes it is indispensable that the actual text can be recovered from the corpus. In our case, therefore, some adaptations were necessary. Most of these could be carried out by a computer program especially written for this purpose, but some correction by hand proved inevitable.

The last step was to adapt the corpus to a format suitable for the search program, which is our principal means of access to the corpus. To this end, the corpus was devided into a search corpus and a text-forms corpus. The search corpus was to contain pairs of lemmata and morphological codes. The text-forms were gathered into sentences and given a header consisting of a sentence number and a text-reference. The reason for omitting text-forms from the search corpus was to speed up searching and, for almost every practical purpose, the redundancy of

these forms: let us suppose, for example, that someone wants to collect instances of the form IOVEM. Instead of searching for this form, one can also direct the program to look for combinations of the lemma IVPPITER and the code for an accusative singular noun[1].

The text-forms file is accessed only when the search program encounters an instance of the pattern to be found and is requested to display the sentence in which it occurs.

An example of a session with this search program will follow below.

## 2. The search program

In view of the type of information that is contained in the corpus, it can be consulted in those cases in which questions can be stated in lexical or categorial terms. Our means of formulating these questions is a pattern-matching program. This program, called "Query", was developed at the Faculty of Arts computer department at the University of Amsterdam by G.J. van der Steen. Here we cannot describe at length how this program operates [for detail information, see Van der Steen (1982)], but, by means of an example, we shall illustrate how it can assist the linguist in conducting research.

The problem to be solved concerns the word order within the Latin ablative absolute construction, more specificially the relative order of Subject (usually a noun or noun phrase) and Predicate (usually a participle) in this construction.

As a first step the linguist may want to collect instances of ablative absolute constructions indiscriminately. We suppose that he has chosen Caesar's *Bellum Gallicum* to be the corpus on which to base his study. The search program is directed to gather all sentences containing an ablative absolute

1. One would need the actual text-forms only if one would like to distinguish between e.g. *-es* and *-is* forms in the accusative plural.

construction. After ca. 75 seconds[1] it will have scanned the whole of the Bellum Gallicum and be ready to display the instances it has found. The first part of the output runs as follows :

```
* [#*********AD]


------------------ CORPUS A ----------------------

10  -  CAES,BG,1:2,1 is Marco Messala Marco Pisone
consulibus regni cupiditate inductus coniurationem
nobilitatis fecit et civitati persuasit ut de finibus suis
cum omnibus copiis exirent perfacile esse cum virtute
omnibus praestarent totius Galliae imperio potiri.
20  -  CAES,BG,1:3,8 hac oratione adducti inter se fidem et
ius iurandum dant et regno occupato per tres potentissimos
ac firmissimos populos totius Galliae sese potiri posse
sperant.
26  -  CAES,BG,1:5,2 ubi iam se ad eam rem paratos esse
arbitrari sunt oppida sua omnia numero ad duodecim vicos ad
quadringentos reliqua privata aedificia incendunt frumentum
omne praeter quod se cum portaturi erant comburunt ut domum
reditionis spe sublata paratiores ad omnia pericula
subeunda essent trium mensum molita cibaria sibi quemque
domo efferre iubent.
27  -  CAES,BG,1:5,4 persuadent Rauracis et Tulingis et
Latobrigis finitimis uti eodem usi consilio oppidis suis
vicis que exustis una cum iis proficiscantur Boios que qui
trans Rhenum incoluerant et in Agrum Noricum transierant
Noreiam que oppugnabant receptos ad se socios sibi
adsciscunt.
32  -  CAES,BG,1:6,4 omnibus rebus ad perfectionem
comparatis diem dicunt qua die ad ripam Rhodani omnes
conveniant.
33  -  CAES,BG,1:6,4 is dies erat ante- diem- v kalendas-
apriles- Lucio Pisone Aulo Gabinio consulibus.
```

The first line displayed is the pattern which was presented to the program. It means "select any sentence containing a sequence of any lemma (this is expressed by the asterisk, the "dont'care" symbol: *) followed by a code containing the letters AD in 10th and 11th position (the code for ablative absolute); positions 1 through 9 may contain any character (this is also expressed by a "don't care" symbol).

1. On a Data General Eclipse S/200 mini-computer. The program is presently being adapted to run on a newly installed Eclipse MV/4000.

After studying, let us say, a hundred instances of these constructions, the researcher finds that the order Subject – Predicate predominates (actually, all instances given above have this order). Nevertheless, the reversed order is not rare. The ratio is about 65 % Subject – Predicate order, 35 % the reversed order. The word order seems to depend, to some extent, on the meaning of the Predicate: if this is a participle of a semantically rather empty verb, such has *habeo* or *facio*, the order Subject – Predicate seems to be almost obligatory. Instead of going through all thousand cases of ablative absolute constructions in Caesar in order to test this hypothesis, one can instruct the search program to select all instances of these verbs within ablative absolute constructions. The output would begin as follows :

```
FACIO [#*********AD] .OR. HABEO [#*********AD]


------------------ CORPUS A ----------------------

42  -  CAES,BG,1:8,4 Helvetii ea spe deiecti navibus
iunctis ratibus que compluribus factis alii vadis Rhodani
qua minima altitudo fluminis erat nonnumquam interdiu
saepius noctu si perrumpere possent conati operis munitione
et militum concursu et telis repulsi hoc conatu
destiterunt.
66  -  CAES,BG,1:13,1 hoc proelio facto reliquas copias
Helvetiorum ut consequi posset pontem in Arari faciendum
curat atque ita exercitum traducit.
131  -  CAES,BG,1:24,4 Helvetii cum omnibus suis carris
secuti impedimenta in unum locum contulerunt ipsi
confertissima acie reiecto nostro equitatu phalange facta
sub primam nostram aciem successerunt.
160  -  CAES,BG,1:29,3 eorum qui domum redierunt censu
habito ut Caesar imperaverat repertus est numerus milium
centum et decem.
180  -  CAES,BG,1:32,1 hac oratione ab Diviciaco habita
omnes qui aderant magno fletu auxilium a Caesare petere
coeperunt.
```

The search pattern appearing at the head of the output shows another feature of the Query program : the use of logical operators within patterns[1]. Here the program is directed to select all sentences containing either *facio* or *habeo*, provided

1. Other operators are "NOT" (requiring that the sentence in order to match may not contain the specified pattern; and "-", the "don't line" symbol between two patterns, admitting any number of words (including zero) to intervene between the leftmost and the rightmost pattern.

that they are followed by the code for ablative absolute. As it turns out, the order is almost exclusively Subject – Predicate[1].

Another factor wich appears to be relevant for the word order is the presence of the pronoun *hic* within the Subject noun phrase. Especially when the pronoun has a sentence connecting function, the phrase shows a strong tendency to take the first place within the ablative absolute construction. Such cases, however, are more difficult to select than the ones referred to above. The program must search for instances of the ablative absolute coocurring with the pronoun *hic* (in the ablative), but there is no guarantee that the pronoun will actually be a constituent of the ablative absolute phrase. Consider the following test :

```
( HIC [#*45F] .AND. * [#***F*****AD] ) .OR. ^
( HIC [#*450] .AND. * [#***O*****AD] )
```

```
------------------ CORPUS A ----------------------

20  -  CAES,BG,1:3,8 hac oratione adducti inter se fidem et
ius iurandum dant et regno occupato per tres potentissimos
ac firmissimos populos totius Galliae sese potiri posse
sperant.
51  -  CAES,BG,1:10,5 compluribus his proeliis pulsis ab
Ocelo quod est citerioris provinciae extremum in fines
Vocontiorum ulterioris provinciae die septimo pervenit inde
in Allobrogum fines ab Allobrogibus in Segusiavos exercitum
ducit.
66  -  CAES,BG,1:13,1 hoc proelio facto reliquas copias
Helvetiorum ut consequi posset pontem in Arari faciendum
curat atque ita exercitum traducit.
80  -  CAES,BG,1:14,7 hoc responso dato discessit.
91  -  CAES,BG,1:16,5 ubi se diutius duci intellexit et
diem instare quo die  frumentum militibus metiri oporteret
convocatis eorum principibus quorum magnam copiam in
castris habebat in his Diviciaco et Lisco qui summo
magistratui praeerat quem vergobretum appellant Haedui qui
creatur annuus et vitae necis que in suos habet potestatem
graviter eos accusat quod cum neque emi neque ex agris sumi
posset tam necessario tempore tam propinquis hostibus ab
iis non sublevetur praesertim cum magna ex parte eorum
precibus adductus bellum susceperit.
161  -  CAES,BG,1:30,1 bello Helvetiorum confecto totius
fere Galliae legati principes civitatum ad Caesarem
gratulatum convenerunt intellegere sese tametsi pro
```

1. For more details see De Jong (1982).

```
veteribus Helvetiorum iniuriis populi Romani ab his poenas
bello repetisset tamen eam rem non minus ex usu terrae
Galliae quam populi Romani accidisse propterea quod eo
consilio florentissimis rebus domos suas Helvetii
reliquissent uti toti Galliae bellum inferrent imperio que
potirentur locum que domicilio ex magna copia deligerent
quem ex omni Gallia opportunissimum ac fructuosissimum
iudicassent reliquas que civitates stipendiarias haberent.
180  -  CAES,BG,1:32,1 hac oratione ab Diviciaco habita
omnes qui aderant magno fletu auxilium a Caesare petere
coeperunt.
```

The program is directed to search for sentences in which both the pronoun (code 45) *hic* and a word serving as a Predicate within an ablative absolute construction (code AD), both in the ablative singular (code F) *or* for sentences containing a similar pattern, but in this case with ablative plurals (code 0)[1].

The output of the program in this case contains many irrelevant instances, for example in the first (*hac ... regno occupato*), fifth (*convocatis ... his*) and sixth instance (*his ... florentissimis*). In these cases there is no direct syntactic relation between the pronoun and the ablative absolute.

Another approach would be to put a limit on the distance between *hic* and the ablative absolute, as in the following example :

```
HIC [#*45F] * [#***F*****AD] .OR. ^
HIC [#*450] * [#***0*****AD] .OR. ^
* [#***F*****AD] HIC [#*45F] .OR. ^
* [#***0*****AD] HIC [#*450] .OR. ^
HIC [#*45F] * % * [#***F*****AD] .OR. ^
HIC [#*450] * % * [#***0*****AD] .OR. ^
* [#***F*****AD] * % HIC [#*45F] .OR. ^
* [#***0*****AD] * % HIC [#*450]
```

```
------------------ CORPUS A ----------------------

51  -  CAES,BG,1:10,5 compluribus his proeliis pulsis ab
Ocelo quod est citerioris provinciae extremum in fines
Vocontiorum ulterioris provinciae die septimo pervenit inde
in Allobrogum fines ab Allobrogibus in Segusiavos exercitum
ducit.
```

1. As one can see, the codings for case and number have been collapsed into one symbol. Since for most applications this is somewhat unpractical (though not so in the present example), the corpus will be recoded with separate codes for case and number.

```
66  -  CAES,BG,1:13,1 hoc proelio facto reliquas copias
Helvetiorum ut consequi posset pontem in Arari faciendum
curat atque ita exercitum traducit.
80  -  CAES,BG,1:14,7 hoc responso dato discessit.
185  -  CAES,BG,1:33,1 his rebus cognitis Caesar Gallorum
animos verbis confirmavit pollicitus que est sibi eam rem
curae futuram magnam se habere spem et beneficio suo et
auctoritate adductum Ariovistum finem iniuriis facturum
186  -  CAES,BG,1:33,2 hac oratione habita concilium
dimisit.
236  -  CAES,BG,1:41,1 hac oratione habita mirum in modum
conversae sunt omnium mentes summa que alacritas et
cupiditas belli gerendi iniecta est princeps que decima
legio per tribunos militum ei gratias egit quod de se
optimum iudicium fecisset se que esse ad bellum gerendum
paratissimam confirmavit.
```

The result of this search-command is that not more than one word is allowed to appear between the pronoun and the predicate of the ablative absolute. Thus the number of irrelevant instances is drastically reduced: the instances that were found during the previous search are now skipped. At a cost, however. Some cases which are relevant are also skipped, such as the seventh instance of the previous search. Moreover, the matching pattern becomes increasingly complex and therefore more difficult to state without making errors.

Considerations such as these have induced us to look for ways of marking synctactic boundaries within sentences. By means of these boundaries one would be able to delimit the domain within which the program may search for cooccurring elements. In the next section, we shall discuss this further.

In this section, we have seen that a search program has some major advantages over conventional means of access to a corpus, such as lexica : one may, for example, search for cooccurring items or for non-occurrences of some particular pattern. These advantages are particularly evident when one uses a lemmatized and morphologically coded corpus, such as the LASLA corpus. Here, one is able to search for lexical and morphological categories as such, so that a linguist interested in the functions of, for example, the imperfect tense or the genitive case may find instances of these categories, whatever the lemma, with or without specifying a particular environment.

## 3. Furnishing the LASLA corpus with syntactic information

As we have seen, it is usually impossible to define the domain in which a search pattern is to be found. In some cases an extra restriction on the match pattern can be a help. The solution offered above consisted of reducing the number of words appearing between two elements of a pattern. Another way to proceed can be to demand that no conjunction should intervene between the words that match, nor a verb belonging to a subordinate clause. Strategies like the latter are based on the fact that many subordinate clauses in Latin are headed by the conjunction or relative pronoun and end with the verb : conjunction and verb serve more or less as 'natural' opening and closing parentheses. Apart from the fact that heuristics like these tend to eliminate a large number of useful instances, the second solution has a disadvantage that it cannot always cope with nested subordinate clauses.

Such considerations made us decide to add more syntactic information to the LASLA corpus. At the same time, the computer department of the Faculty of Arts was working on a more advanced search program, in which our extensions could be handled.

Since automatic parsing of natural languages still poses large problems, even when applied to coded texts like the LASLA corpus, we decided to proceed manually, at least in the first stage. As a start, we limited ourselves to putting parentheses around finite clauses. This way we expected to be able to process a considerable amount of text. Moreover, it was most often absence of boundaries between subordinate clauses that caused problems when searching in the corpus. Also, we hoped that practical experience would prove to be useful in a more advanced future project, where more syntactic categories would be distinguished.

In the course of the project until now, a few problems arose. One of the problems we did not solve is syntactic discontinuity ('hyperbaton').

We define discontinuity as the phenomenon that occurs when two or more words that belong to the same clause or phrase are separated by words belonging to another clause or phrase which is not directly subordinate to the separated clause or phrase. As an example, consider :

(1) Boios petentibus Haeduis, quod egregia virtute erant cogniti, ut in finibus suis conlocarent concessit. (Caes. BG 1,28,5)

Here *Boios* syntactically belongs in the clause *ut ... concessit,* but it separated from it by a 'higher' clause, *petentibus Haeduis.*

In this stage of the project, sentences such as these are not analyzed into separate clauses. They will be dealt with when a procedure has been developed to cope with discontinuity (see below).

Another problem which confronted us concerns ellipsis, occurring most frequently, though not exclusively, in coordinated structures. In many cases we had to decide whether these structures could still be viewed as separate finite clauses and consequently be parenthesized as such. Consider the following examples :

(2) Saepe clamore ipso militum aut instructione aspectuque signorum magnas copias pulsas esse et vidimus et audivimus. (Cic. Caec. 43)
(3) Gallos ab Aquitanis Garunna flumen, a Belgis Matrona et Sequana dividit. (Caes. BG 1,1,2)
(4) Triumphavit L. Sulla, triumphavit L. Murena de Mithridate (Cic. Man. 8).
(5) Helvetii iam .. suas copias traduxerant et in Aeduorum fines pervenerant (Caes. BG 1,11,1).

A case such as (2) one would most likely want to analyse as an instance of verbal coordination rather than to distinguish two separate clauses, the second of which underwent ellipsis of *saepe* and the infinitival construction. On the other hand, in (5) a two-clause analysis is much more plausible, whereas cases such as (3) and (4) are somewhere in between. We emphasize that we do not want to involve ourselves in the theoretical debate on coordination (concerning, e.g. 'conjunction reduction' and competing analyses), but simply want to fix intuitively plausible boundaries that may serve to define search domains on the level of the finite clause. As a practical criterium we use the number of ellipted phrases. Roughly: when more than one phrase has been ellipted, we do not analyze the clause into separate coordinates. Thus, (2) and (3) are not analyzed into two distinct clauses, while (4) and (5) are.

When parentheses for clause demarcations have been inserted, (5) will have the following form :

```
(5') ( ( Helvetii iam .. suas copias traduxerant ) ( et in
     Aedorum fines pervenerant ) ) (Caes. BG 1,11,1).
```

Notice that the coordinator *et* has been included within the clause. This, again, is a theoretical issue, but our decision has been taken for practical reasons. The enclitic coordinator *-que* would otherwise have caused a great deal of discontinuity.

As an example of a complete analysis (on the clause level), consider the following instance (The layout is the same as displayed by the interactive coding program, to make nesting transparent).

```
(6) ( nunc
      ( cum et auctoritatis in me tantum sit
         ( quantum vos honoribus mandandis esse voluistis
         )
        et ad agendum facultatis tantum
         ( quantum homini vigilanti ex forensi usu prope
           cotidiana dicendi exercitatio potuit adferre
      ) )
     certe et
      ( si quid auctoritatis in me est
      )
     apud eos utar
      ( qui eam mihi dederunt
      )
     et
      ( si quid in dicendo consequi possum
      )
     eis ostendam potissimum
      ( qui ei quoque rei fructum suo iudicio tribuendum
        esse duxerunt
   ) ) (Cic. Man. 2)
```

When the search program is processing such a sentence (in the actual search corpus not consisting of text-forms, but of combinations of lemmata plus code), it can be requested to stick to one level at a time : therefore, when searching for a dative pronoun at the main clause level (*nunc ... certe et ... apud eos utar et ... eis ostendam potissimum ...*), it will only find *eis* and not *mihi* in one of the intervening subordinate clauses.

## *Future perspectives*

Sofar our efforts in marking synctactic boundaries have been rather limited. We confined ourselves to finite clauses, leaving aside non-finite ones (such as accusativus cum infinitivo, ablativus absolutus) and noun phrases. The main reason for

doing so was the consideration that in Latin discontinuity in word order does not occur as frequently in main and subordinate clauses as it does with lower level constituents.

Ideally, in a corpus provided with syntactic boundaries smaller constituents at all levels should be demarcated. In our view, the structure of such a corpus ought to be as follows :
1. Each syntactic constituent is enclosed in parentheses.
2. Opening parentheses have a label that marks the function of a constituent, for instance 'SUBJ:(pater)' for the Subject phrase *pater*.
3. Discontinuities and ellipsis are marked by a special formalism.

The production of this type of corpus can be facilitated by having part of the work done by the computer. This seems to be an unrealistic goal when the LASLA corpus in its current form is to be the input of a program with that task. Therefore, we think the following manual preprocessing actions will be necessary :
1. Marking clausal boundaries (as it is done in the current project).
2. Marking discontinuity and ellipsis.
3. Disambiguating 'difficult constructions' i.e. syntactic constructions which the program is unable to analyze.

As far as we know, no solution of the problem concerning discontinuity has been proposed as yet. It seems useful, therefore, to present our own view on how this difficulty should be overcome.

Generally, discontinuous phrases should be parenthesised in the same way as continuous ones. When putting labeled parentheses around discontinuous phrases, one should proceed as if one were dealing with a continuous one. However, at the position where discontinuity starts, special action has to be taken. Consider for example the sentence

(7) Orsa procella est.

The verb-phrase *orsa + est* is discontinuous because of the intervening noun *procella*. The first part of the Predicate (*orsa*) is labeled with the same label[1] as the second part (*est*).

1. The names of labels are given here only as examples.

Thus, the resulting sentence is :

(7') S:(PRED:(Orsa) SUBJ:(HEAD:(procella)) PRED:( est))

(Later we shall see how both parts can be linked together).

Sentence (8) is a more complicated example :

(8) Magna orsa procella est.

Here we are confronted with a noun phrase that has the hierarchical structure SUBJ:(HEAD:(procella) MODIF:(magna)). In order to maintain the hierarchy, the label is repeated in both parts :

(8') S:(SUBJ:(MODIF:(Magna)) PRED:(orsa)
SUBJ:(HEAD:(procella)) PRED:(est))

In order to create a link between the separate parts of discontinuous groups, all labels (also those of continuous groups) should be extended with a unique reference number. All parts of discontinuous groups will receive the same reference. Sentence (8) will then have the following form :

(8") S_1:(SUBJ_2:(MODIF_3:(Magna)) PRED_4:(orsa)
SUBJ_2:(HEAD_5:(procella)) PRED_4:(est))

For the sake of convenience, a special marker is added to the label with reference, which will be used to indicate whether the group is continuous or not, and in the case of discontinuity, whether it is the first, next or last part of the group.

In practice, we think the creation of this corpus will consist of the following steps :

1. Defining a standard word order for all syntactic groups to be distinguished. This standard word order, which should not allow discontinuities, must be such that it can be easily analyzed by a computer program.
2. Arranging by means of a special-purpose interactive program the words of a given sentence in such a way that they comply with the designed standard word order; information about the original word order is to be maintained. In this phase additional information can, if need be, be provided to resolve syntactic ambiguities.
3. Parsing the sentence thus adapted automatically (the computer department has already developed a parser generating program) and having labeled parentheses added correctly by the parser.

4. After a succesful analysis of the sentence, a special program should rearrange the words into their original order and add references to the labels that have been generated. In this phase, discontinuities can be coded automatically, in accordance with the rules sketched above.

The problem of ellipsis has up to now received less of our attention than that of discontinuity; consequently, our ideas concerning the former are not as well developed as those concerning the latter[1].

It goes without saying that a search program should be able to comply with the demands posed by the extensions added to the corpus. One will need :
1. A way to handle parentheses (especially nested ones).
2. Special don't care symbols for one or any number of opening and closing parentheses.
3. A method of relating don't care symbols to each other by means of ab identifier (demanding, for instance, that the next occurence of a don't care identifier matches with the same (sub-)string as an earlier one). This is, of course, essential to our method which uses reference numbers in the case of discontinuous syntactic groups.

A program that meets these needs is being developed at the computer department of the Faculty of Arts.

## 4. Concluding remarks

In the foregoing we have shown how the LASLA corpus has been converted into a tool for (sentence-based) linguistic analysis. It owes its usefulness in particular to its categorial and morphological codes, wich makes it unique among Latin text corpora.

Adding syntactic information to a corpus is a complicated process. We have described the steps that have already been taken in that direction, and how we intend to proceed from there.

---

1. A possible solution might be to repeat the label with reference number of the ellipted constituent on the position where is left out.

## References

De Jong, J.R., (1982). Word order within the Latin Ablative Absolute construction. In : S. Daadler and M. Gerritsen (eds.), *Linguistics in the Netherlands*, 1982.

Van der Steen, G.J., (1982). A treatment of queries in large text corpora. In : S. Johansson (ed.), *Computer corpora in English language research*.