

SHALLOW PARSING AND FUNCTIONAL STRUCTURE IN ITALIAN CORPORA

Rodolfo Delmonte

Ca' Garzoni-Moro, San Marco 3417

Università "Ca Foscari"

30124 - VENEZIA

Tel. 39-41-2578464/52/19

E-mail: delmont@unive.it

Website: <http://byron.cgm.unive.it>

ABSTRACT

In this paper we argue in favour of an integration between statistically and syntactically based parsing by presenting data from a study of a 500,000 word corpus of Italian. Most papers present approaches on tagging which are statistically based. None of the statistically based analyses, however, produce an accuracy level comparable to the one obtained by means of linguistic rules [1]. Of course their data are strictly referred to English, with the exception of [2, 3, 4]. As to Italian, we argue that purely statistically based approaches are inefficient basically due to great sparsity of tag distribution – 50% or less of unambiguous tags when punctuation is subtracted from the total count. In addition, the level of homography is also very high: readings per word are 1.7 compared to 1.07 computed for English by [2] with a similar tagset.

The current work includes a syntactic shallow parser and a ATN-like grammatical function assigner that automatically classifies previously manually verified tagged corpora. In a preliminary experiment we made with automatic tagger, we obtained 99,97% accuracy in the training set and 99,03% in the test set using combined approaches: data derived from statistical tagging is well below 95% even when referred to the training set, and the same applies to syntactic tagging. As to the shallow parser and GF-assigner we shall report on a first preliminary experiment on a manually verified subset made of 10,000 words.

1. INTRODUCTION

We assume, together with [1] that POS tagging is essentially a syntactically-based phenomenon and that by cleverly coupling stochastic and linguistic processing one should be able to remedy some if not all of the drawbacks usually associated with the two approaches, when used in isolation. However, as will be shown in detail in the following section, rather than using FSA we use an RTN both for training and for parsing. As to the statistical part, we don't use HMMs but only conditional probabilities on the bases of trigram information as discussed below.

Syntactic shallow parsing is accomplished in our case by using the same RTN we use for tag disambiguation. It is made up of 1700 arcs and 24 nets, and we use it in a non-recursive way, as explained below. Data for the construction of the RTN were derived from the manual annotation of 40,000 token corpus suite which is then used as test set. The manual annotators used 44 broad syntactic tags, 10 of which main constituents or nonterminals, and 34 preterminals. Frequency of occurrence associated to each rewrite rule is used as organizing criteria in the ordering of the arcs contained in each node of each net.

The task of the Shallow Parser is that of creating syntactic structures which are eligible for Grammatical Function assignment. This task is made simpler given the fact that the disambiguator will associate a net/constituency label to each disambiguated tag. Parsing can then be defined as a Bottom-Up collection of constituents which contain either the same label, or which may be contained in/be member of the same net/higher constituent. No attachment is performed in order to avoid being committed to structural

decisions which might then reveal themselves to be wrong. We prefer to perform some readjustment operations after structures have been built rather than introducing errors from the start. Readjustment operations are in line with LFG theoretical framework which assumes that f-structures may be recursively constituted by subsidiary f-structures, i.e. by complements or adjuncts of a governing predicate. Restructuring is executed taking advantage of agreement information which in languages like Italian, i.e. in morphologically rich languages, can be fruitfully used to that aim.

Syntactic structure is derived from shallow structures by a restricted and simple set of rewriting operations which are of two categories: deletions, and restructuring.

a. Deletion

Delete structure internally the same constituent label that appears at the beginning as in Noun Phrases.

b. Restructuring

As explained above, we want to follow a policy of noncommittal as to attachment of constituents: nonetheless, there are a number of restructuring operations which can be safely executed in order to simplify the output without running the risk of taking decisions which shall have later to be modified. We perform the following restructuring operations:

- i. integrate sn/NP into sp/PP heads;
- ii. integrate predicative sa/AP into adjacent preceding sn/NP;
- iii. integrate sv2/VPinfinitival into sp/PP heads;
- iv. integrate spd/PPof into governing sn/NP

So the basic task of the shallow parser is that of building shallow structures for each safely recognizable constituent and then pass this information to the following modules. Agreement structures are used to transform shallow

syntactic structures into annotated f-structures. In turn, this information is used in an ATN-like fashion to produce the final F-structure associated to that sentence. Remember that in LFG f-structures may be atomic or they may contain other f-structures: typically, a SUBJECT f-structure will contain its PRED and f-features but also its MODifiers, a POSSESSIVE, its ADJUNCTS etc. So the idea is to allow for OF-PPs and MODs to be included in the same F-structure associated to a given NP. Complements and adjuncts of the verbal predicate are all at the same level; in case they are sentential ones they will be marked off by a different sentence index. The intermediary structure is the Agreement structure which is used extensively to check for Grammatical Function association process.

2. SYNTACTIC CONSTITUENCY ANNOTATION

The first problem to be solved when starting work on a corpus in order to produce a syntactic structure annotation, is the choice of representation, or the syntactic annotation scheme. As with tagging, the scheme must be consistent, it could be used as gold standard for parser testing or as a basis for the induction of stochastic grammars and lexical representations. In our case then, syntactic constituency should be used as a first level of syntactic and semantic representation, with the declared aim of translating main constituent labels into functional labels, as a second step of the overall project. As a further effort and a short term work in progress, we have already been developing tools for predicate-argument extraction with the help of subcategorization frames available in our lexicon of Italian and documented in [10].

The main sources of information in the field of syntactic annotation scheme are related to the Penn Treebank (hence PT) [12, 13], which is remarkable as to extension of the coverage and documentation of linguistic phenomena. However, we do not share the theory underlying its linguistic representation: as a result, much of the bracketing is non comparable. The PT has chosen the so called first generation generative linguistics as a theoretical standard which is a sound and consistent choice. However, they then decided to enrich syntactic constituency with functional labels and also introduced quite a number of non standard additional labels which increased the overall number of

constituents but reduced its perspicuity. As a result, they use 22 symbols for main constituent and 32 more for functional annotation. We also use 22 symbols for syntactic constituency but they are different from the PT's ones.

The inventory we use is much smaller and follows the basic intuitions of the XBAR syntax, while having as its main goal that to serve as an interface as simple as possible to the following levels of representations: the functional and the semantic ones. In NPs we let the elements of the specifier be at the same level of the head Noun. In sum, we allow for Head/Complement convention, the Head being the last element on the right in Italian. We use specialized constituent names for a number of complements or adjuncts, in order to help the work of the functional annotation tool.

The PT uses a generativist constituency which is related to chomskian syntax of the '60s/'70s. In particular, they use Chomsky-adjunction to mark the presence of an argument or an adjunct. Seen that they only have one layer of syntactic representation, whereas we allow for two, they include all semantic information at constituent level. In particular, they introduce all possible empty categories in the syntactic constituents with coindexation. In case of discontinuous or non canonical order of constituents, they use special constituent names, like SINV (Inverted Sentence), to allow for the subject NP to be automatically recovered. As an example we report the bracketing for "John's decision to leave":

```
(NP (NP John 's)
      decision
      (S (NP-SBJ *)
          (VP to
            (VP leave))))
```

compared to the Italian, "la decisione di Gino di partire"
SN-[la-art, decisione-n,

```
SPD-[di-pd, SN-[Gino-nh] ]
SV2-[di-pt, partire-viin] ]
```

where we can see that the level of embedding in PT is 4 brackets, whereas it is 2 brackets in our representation. We also include tags and use Prolog lists and terms, whereas they use Lisp formalism.

We report here below the list of constituents in our representation for Italian corpora.

TABLE 4. List of Syntactic Constituents and their meaning

F	sentence, starting with subject SN or SV2; or in case subject is missing starting with IBAR
SN	noun phrase, including its complements and/or adjuncts
SA	adjectival phrase, including its complements and/or adjuncts
SP	prepositional phrase
SPD	prepositional phrase DI / "of"
SPDA	prepositional phrase DA / "by,from"
SAVV	adverbial phrase, including its complements and/or adjuncts
IBAR	verbal nucleus with finite tense and all adjoined elements like clitics, adverbs and negation

SV2	F for infinitival clause
SV3	F for participial clause
SV5	F for gerundive clause
FAC	CP for sentential complement
FC	CP for Coordinate sentences (also ellipsed and gapped)
FS	CP for Subordinate sentence
FINT	CP for +wh interrogative sentence
FP	CP for punctuation marked parenthetical or appositional sentence
F2	CP for relative clause
CP	Generically for dislocated or fronted, sentential adjuncts
COORD	Coordination with coordinating conjunction as head
COMPT	Transitive/Passive/Ergative/Reflexive Complement
COMPIN	Intransitive/Unaccusative Complement
COMPC	Copulative/Predicative Complement

From the point of view of their relations to grammatical functional (GFs) labeling, and their organization into adjuncts and arguments, syntactic constituents are divided up into two main subgroups: functional constituents and substantial constituents. This subdivision reflects theoretical assumptions which are derived from LFG theory. In particular, we shall have that FUNCTIONAL

CONSTITUENTS are internally divided up into Structural and Lexical Functional Constituents. Structural constituents are used to set complements apart and map them into the appropriate GFs. In addition, they contain F and CP where F has the task of indicating the canonical sentential constituent and CP when present, indicates the presence of sentential adjuncts, or some discontinuity in the utterance.

Table 5. TABLE OF SYNTACTIC CONSTITUENTS

STRUCTURAL	LEXICAL		SUBSTANTIAL
F	FAC	SP	SN
CP	FC	SPD	SA
COMPT	FS	SPDA	SAVV
COMPIN	FINT		IBAR
COMPC	FP		SV2
	F2		SV3
	COORD		SV5

3. THE SHALLOW PARSER

The task of the Shallow Parser is that of creating syntactic structures which are eligible for Grammatical Function assignment. This task is made simpler given the fact that the disambiguator will associate an elementary tree label to each disambiguated tag. Parsing can then be defined as a Bottom-Up collection of constituents which contain either the same label, or which may be contained in/be member of the same net/higher constituent. No attachment is performed in order to avoid being committed to structural decisions which might then reveal themselves to be wrong. We prefer to perform some readjustment operations after structures have been built rather than introducing errors from the start. Readjustment operations are in line with

LFG theoretical framework which assumes that f-structures may be recursively constituted by subsidiary f-structures, i.e. by complements or adjuncts of a governing predicate. So the basic task of the shallow parser is that of building shallow structures for each safely recognizable constituent and then pass this information to the following modules. Here below we present the output of each parsing module starting from the first output and then showing the final output with Grammatical Functions. The comment on the algorithm follow each structural representation. We start by giving the syntactic shallow structure associated to the sentence portion we chose as an example:

“le esperienze delle città estere ribadiscono l’importanza delle infrastrutture nel creare in un’area un ambiente migliore” / the experiences of foreign towns reassert the

importance of the infrastructures in creating in an area a better environment

```
[f,
[ sn, art-Le, sn, n-esperienze ],
[ spd, partd-delle ],
[ sn, n-città ],
[ sa, ag-estere ],
[ ibar, vt-ribadiscono ],
[ sn, art-l, sn, n-importanza ],
[ spd, partd-delle ],
[ sn, n-infrastrutture ],
[ sp, part-nel ],
[ sv2, vit-creare ],
[ sp, p-in ],
[ sn, art-un_ n-area ],
[ sn, art-l, sn, n-ambiente ],
[ sa, ag-migliore ]
]
```

4. SYNTACTIC READJUSTMENT RULES

Syntactic structure is derived from shallow structures by a restricted and simple set of rewriting operations which are of two categories: deletions, and restructuring.

a. Deletion

Delete structure internally the same constituent label that appears at the beginning as in Noun Phrases,

```
[ sn, art-l, sn, n-importanza ] => [ sn, art-l, n-importanza ]
```

b. Restructuring

As explained above, we want to follow a policy of noncommittal as to attachment of constituents: nonetheless, there are a number of restructuring operations which can be safely executed in order to simplify the output without running the risk of taking decisions which shall have later to be modified. We perform the following restructuring operations:

- i. integrate sn/NP into sp/PP heads;
- ii. integrate predicative sa/AP into adjacent preceding sn/NP;
- iii. integrate sv2/VP into sp/PP heads;
- iv. integrate spd/PPof into governing head sn/NP

```
[f,
[ sn, art-Le, n-esperienze,
  [ spd, partd-delle, [ sn, n-città, [ sa, ag-estere ] ] ] ]
[ ibar, vt-ribadiscono ]
[ sn, art-l, n-importanza,
  [ spd, partd-delle, [ sn, n-infrastrutture ] ] ]
[ sp, part-nel, [ sv2, vit-creare ] ],
[ sp, p-in, [ sn, art-un_ n-area ] ],
[ sn, art-l, n-ambiente, [ sa, ag-migliore ] ]
]
```

Restructuring is executed taking advantage of agreement information which in languages like Italian, i.e. in morphologically rich languages, can be fruitfully used to that aim. Here below are agreement structure for the sentence excerpt under analysis. As can easily be noticed, matching operations are simplified by the agreement shortened labels which are extracted for each wordform and proposed to the feature list.

sn-f_p-[w=le, def=def, gen=f, num=p] sn-f_p-[w=esperienze, root=esperienz, type=com, gen=f, num=p, head=com] spd-di-[w=delle, root=di, cat1=prep, p2=il, cat2=art, type=det, gen=f, num=p] sn-f-[w=città, root=città, type=invar, gen=f, head=invar]
sa-f_p-[w=estere, root=ester, gen=f, num=p] subj-p_3-[w=ribadiscono, root=ribad, scat=[intr, tr], mood=indic, tense=pres, pers=3, num=p] sn-m_s-[w=l, def=def, num=s] sn-f_s-[w=importanza, root=importanz, type=com, gen=f, num=s, head=com] spd-di-[w=delle, root=di, cat1=prep, p2=il, cat2=art, type=det, gen=f, num=p] sn-f_p-[w=infrastrutture, root=infrastruttur, type=com, gen=f, num=p, head=com] sp-in-[w=nel, root=in, cat1=prep, p2=il, cat2=art, type=det, gen=m, num=s] sv2-pPro-[w=creare, root=cre, mood=inf, tense=pres, scat=tr] sp-in-[w=in, root=in, cat=prep] sn-f_s-[w=un, def=ind, gen=f, num=s] sn-f_s-[w=area, root=are, type=com, gen=f, num=s, head=com] sn-m_s-[w=l, def=def, num=s] sn-m_s-[w=ambiente, root=ambient, type=com, gen=m, num=s, head=com] sa-mf_s-[w=migliore, root=miglior, gen=mf, num=s]

The syntactic structure produced at the end of the agreement structure building process is the following:

f-[
 sn-[art-Le, n-esperienze,
 spd-[partd-delle, sn-[n-città, sa-[ag-estere]]]]
 ibar-[vt-ribadiscono]
 sn-[art-l, n-importanza,
 spd[partd-delle, sn-[n-infrastrutture]]]
 sp-[part-nel,
 sv2-[vit-creare,
 sp[p-in, sn-[art-un_, n-area]],
 sn-[art-l, n-ambiente, sa-[ag-migliore]]]]]

5. MANUAL ANNOTATION AND AUTOMATIC SYNTACTIC TAGGING

As commented above, being language-dependent the tagger needs to be based on an accurate analysis of corpora with

an as broad as possible coverage of genre, style and other social and communicative variables. To answer these needs we built our syntactic shallow parser on the basis of manually annotated texts for 60,000 words chosen from different corpora and satisfying the above-mentioned criteria. The annotation was carried out twelve years ago to be used for a text-to-speech system for Italian (DecTalk Italian version) with unlimited vocabulary [14,15].

We report here below the list of the 10 main constituents or net labels used by the annotators, which are a subset of our current syntactic tagset:

F – sentence; F1 – verbless clause; F2 – interrogative, relative and exclamative clauses; SA – adjectival phrase; SAVV – adverbial phrase; SN – nominal phrase; SP – preposition phrase; SQ – quantifier phrase; SV – verbal phrase; SV2 – infinitival verbal phrase.

Table 6. Net Accessibility Preterminals and their Frequency

NET TAG FREQ			NET TAG FREQ			NET TAG FREQ			NET TAG FREQ		
F	P	235	SAVV	AVV	1479	SP	P	6160	SV	VG	147
F	C	218	SN	A	124	SP	PART	5234	SV	VPP	814
F	C	294	SN	ART	3792	SV	AUSA	244	SV	VSUP	518
F1	P	89	SN	DIM	117	SV	AUSE	363	SV2	P	173
F2	P	209	SN	N	1662	SV	CLIT	388	SV2	PT	529
SA	A	353	SN	PRON	338	SV	NEG	318	SV2	VI	217
SA	Q	239	SN	Q	189	SV	V	656			

We took 15% of the training set, i.e. 10,000 tokens, with texts covering most domains, and ran our experiment of automatic syntactic tagging, with the idea of producing only main constituents in case the attempt at building a complete parse failed. The results can be seen in Appendix I where we present an excerpt from the actual text: the level of ambiguity is such that the bracketing will only produce one of the possible interpretations.

Functional structures are built recursively from syntactic constituency by means of an ATN-like functional parser that takes agreement structures and lexical information related to subcategorization frames in order to unify the input structure with the most appropriate functional description.

Each lexical entry in our computational lexicon is organized as follows:

- 1 predicate;
2. syntactic class;

3. aspectual class;

4. semantic class;

5. the list of arguments contains for each argument:

a. syntactic constituency; b. grammatical function;

c. semantic role;

d. selectional restrictions or a control equation for open functions;

no information at all for propositional

arguments.

The computational lexicon we use is organized as shown here below for the entry "assicurare"/assure,ensure,makesure etc.:

```

pred-[w=assicurano, root=assicur, scat=[rifl, tr], mood=indic, tense=pres, pers=3, num=p]
lex_forms=[
assicur-refl-activ-react-[np/subj1/actor/[rifl, +hum], s_bar/scomp/prop/[subj=subj1/x]],
assicur-refl-activ-react-[np/subj1/actor/[rifl, +hum], vinf/vcomp/prop/di/[subj=subj1]],
assicur-refl_in-activ-react-[np/subj1/actor/[rifl, +hum], pp/obl/malef/contro/[ -ani]],
assicur-tr-accomp-react-[np/subj1/address/[ -ani], np/obj1/theme_aff/[ +hum],
pp/obl/source_info/da/[ -ani]],
assicur-tr-accomp-react-[np/subj1/address/[ -ani], np/obj1/theme_aff/[ +hum],
pp/obl/theme/[su,sopra,attorno]/[ -ani]],
assicur-tr-achiev-react-[np/subj1/address/[ +hum], np/obj1/informtn/[ -ani], pp/obj2/goal/a/[ +hum, +abst]],
assicur-tr-achiev-react-[np/subj1/address/[ +hum], np/obj1/informtn/[ -ani], pp/obl/benef/per/[ -ani]],
assicur-tr-activ-react-[np/subj1/actor/[ +hum], np/obj1/theme_aff/[ +hum], pp/obl/source_info/da/[ -ani]],
assicur-tr-activ-react-[np/subj1/actor/[ +hum], np/obj1/theme_aff/[ +hum],
pp/obl/theme/su_sopra_attorno/[ -ani]],
assicur-tr-activ-react-[np/subj1/actor/[ +hum], s_bar/scomp/prop/[subj=subj1/x], pp/obj2/goal/a/[ +hum]],
assicur-tr-activ-react-[np/subj1/actor/[ +hum], vinf/vcomp/prop/di/[subj=subj1]]]

```

We show here below the output f-structure for the sentence commented above:

```

mood=indic
tense=pres
lex_form=ribad
pred-[w=ribadiscono, root=ribad, scat=[intr, tr], mood=indic, tense=pres,
pers=3, num=p]
subj-[def=def, ind(f0), esperienze-[w=esperienze, root=esperienz,
type=com, gen=f, num=p, head=com],
delle-[w=delle, root=di, cat1=prep, p2=il, cat2=art, type=det,
gen=f, num=p],
città-[w=città, root=città, type=invar, gen=f, head=invar]]
obj-[m_s-[def=def], ind(f0), importanza-[w=importanza, root=importanz, type=com, gen=f, num=s, head=com],
delle-[w=delle, root=di, cat1=prep, p2=il, cat2=art, type=det,
gen=f, num=p],
infrastrutture-[w=infrastrutture, root=infrastruttur, type=com, gen=f, num=p, head=com]]
obj-[f_s-[def=ind], area-[w=area, root=are, type=com, gen=f, num=s, head=com], gov_pred=creare, ind(f1)]
adj-[in-[w=nel, root=in, cat1=prep, p2=il, cat2=art,
type=det, gen=m, num=s], gov_pred=ribad, ind(f0)]
vcomp-[creare-[w=creare, root=cre, mood=inf, tense=pres, scat=tr],
gov_pred=in, ind(f1)]

```

6. REFERENCES

- [1] P. Tapanainen and Voutilainen A.(1994), Tagging accurately - don't guess if you know, *Proc. of ANLP '94*, pp.47-52, Stuttgart, Germany.
- [2] Brants T. & C.Samuelsson(1995), Tagging the Teleman Corpus, in *Proc.10th Nordic Conference of Computational Linguistics*, Helsinki, 1-12.
- [3] Lecomte J.(1998), Le Categoriseur Brill14-JL5 / WinBrill-0.3, INaLF/CNRS,
- [4] Chanod J.P., P.Tapanainen (1995), Tagging French - comparing a statistical and a constraint-based method". *Proc. EACL'95*, pp.149-156.
- [5] Brill E. (1992), A Simple Rule-Based Part of Speech Tagger, in *Proc. 3rd Conf. ANLP*, Trento, 152-155.
- [6] Cutting D., Kupiec J., Pedersen J., Sibun P., (1992), A practical part-of-speech tagger, in *Proc. 3rd Conf. ANLP*, Trento.
- [7] Voutilainen A. and P. Tapanainen,(1993), Ambiguity resolution in a reductionistic parser, in *Sixth Conference of the European Chapter of the ACL*, pp. 394-403. Utrecht.
- [8] Pereira F., R.Wright, (1991), Finite-State Approximation of Phrase Structure Grammars, in *Proc. 29th ACL*, Berkeley, 246-255.
- [9] Delmonte R., E.Pianta(1999), Tag Disambiguation in Italian, in *Proc.Treebanks Workshop ATALA*, Paris, pp.18-25.
- [10] Delmonte R.(to appear)(1999), Tag Disambiguation and Shallow Parsing in Italian, A.Abeillé (ed.)(to appear), *Treebanks & Tagging*, Kluwer, pp.20.

- [11] Marcus M. et al.(1993), Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, Vol.19.
- [12] Bies et al.(1995) Bracketing Guidelines for Treebank II Style Penn Treebank Project, and EAGLES' Preliminary Recommendations for the Syntactic Annotation of Corpora (Version of March 1996).
- [13] Johansson, S., E. Atwell, R. Garside & G. Leech. 1986. *The Tagged LOB Corpus: Users' manual. ICAME*, The Norwegian Computing Centre for the Humanities, Bergen University, Norway.
- [14] Delmonte R. G.A.Mian, G.Tisato(1986), A Grammatical Component for a Text-to-Speech System, *Proceedings of the ICASSP'86*, IEEE, Tokyo, 2407-2410.
- [15] Delmonte R., R.Dolci(1989), Parsing Italian with a Context-Free Recognizer, *Annali di Ca' Foscari XXVIII*, 1-2,123-161.
- [8] Delmonte R., E.Pianta(1996), "IMMORTALE - Analizzatore Morfologico, Tagger e Lemmatizzatore per l'Italiano", in *Atti V Convegno AI*IA*, Napoli, 19-22.

APPENDIX I.

Joseph Gicquel, responsabile del servizio telematico del quotidiano parigino Liberation, racconta: "Abbiamo iniziato a offrire un servizio telematico alla fine dell'83, proponendo due servizi di informazione.

LIST OF DISAMBIGUATED TOKENS

i('Joseph'-npro-sn).
i('Gicquel'-npro-sn).
i(',-punt-fp).
i(responsabile-n-sn).
i(del-partd-spd).
i(servizio-n-sn).
i(telematico-ag-sa).
i(del-partd-spd).
i(quotidiano-n-sn).
i(parigino-ag-sa).
i('Liberation'-npro-sn).
i(',-punt-fp).
i(racconta-vt-ibar).
i(':-dirs-fp).
i('"-par-fp).
i(abbiamo-ausa-ibar).
i(iniziato-vppt-sv3).
i(a-pt-sp).
i(offrire-vit-ibar).
i(un-art-sn).
i(servizio-n-sn).
i(telematico-ag-sa).
i(alla-part-sp).
i(fine-n-sn).
i(dell-partd-spd).
i(83-num-sa).

i(',-punt-fp).
i(proponendo-vgt-ibar).
i(due-num-sn).
i(servizi-n-sn).
i(di-pd-spd).
i(informazione-n-sn).
i(',-punto-cp).

SYNTACTIC CONSTITUENTS STRUCTURE

f-[
sn-[npro-Joseph, npro-Gicquel],
fp-[punt-,],
sn-[n-responsabile,
spd-[partd-del,
sn-[n-servizio,sa-[ag-telematico],
spd-[partd-del, sn-[n-quotidiano,
sa-[ag-parigino],
sn-[npro-Liberation],
fp-[punt-,]]]]]],
ibar-[vt-racconta],
fp-[dirs-:, par-".,],
ibar-[ausa-abbiamo, vppt-iniziato],
compt-[sv2-[pt-a, vit-offrire,
sn-[art-un, n-servizio,
sa-[ag-telematico]],
sp-[part-alla, sn-[n-fine,
spd-[partd-dell, sn-[num-83],
fp-[punt-,]]]]]],
sv5-[vgt-proponendo,
sn-[num-due, n-servizi,
spd-[pd-di, sn-[n-informazione]]]],
f-[punto-.]]