Parsing Overlaps

Rodolfo Delmonte, Antonella Bristot, Luminita Chiran, Ciprian Bacalu, Sara Tonelli
Ca' Garzoni-Moro, San Marco 3417
Università "Ca Foscari"
30124 - VENEZIA
Tel. 39-041-2349464/52/19 - Fax. 39-041-5287683
E-mail: delmont@unive.it - website: project.cgm.unive.it

Abstract

Overlaps constitute a challenge for any system for linguistic representations in that they cannot be treated as a one-dimensional event: in order to take into account the purport of an overlapping stretch of dialogue for the ongoing pragmatics and semantics of discourse, we have devised a new annotation schema which is then fed into the parser and produces a multidimensional non-linear syntactic constituency representation.

This paper will present work carried out on the 60,000 words Italian Spontaneous Speech Corpus called AVIP, under national project API. It will present the linguistic annotation tools used, in particular the parser, to produce syntactic structures of overlapped temporally aligned turns. Then it will concentrate on the syntactic, semantic and prosodic aspects related to this debated issue.

The paper will argue in favour of a joint and thus temporally aligned representation of overlapping material to capture all linguistic information made available by the local context. This will result in a syntactically branching node we call OVL which contains both the overlapper's and the overlappee's material (linguistic or no-linguistic). An extended classification of the phenomenon has shown that overlaps contributes substantially to the interpretation of the local context rather than the other way around.

1. Introduction

This paper presents work carried out at the University of Venice for the creation of tools for the annotation of spoken Italian which allow the user to work in a format fit for the visualization of the results in multilevels representation in commercial browsers. The specific topic of this paper will be the characterization of overlaps along the lines of what has been done in MATE project and other international projects in progress like the MEETING project. In the AVIP/API dialogues the quantity of overlapping speech is very high, as has been reported in the national conference on "Parlato Italiano" – Naples, 13-15 February, 2003. At an international level, even though everybody agrees on the relevance of the phenomenon, there is no universal agreement on its representation from the linguistic point of view, in particular as concerns syntactic structure both at constituent and functional level.

In the last few years, in the field of spoken dialogue corpus annotation, level-specific coding tools gradually emerged - for morphosyntactic annotation, co-reference annotation, dialogue acts annotation etc., as described in the MATE (Multi-level Annotation Tools Engineering) project report on the state of the art in spoken dialogue annotation tools. NITE pursues the same objectives as its predecessor project MATE. The main difference is that NITE goes beyond spoken dialogue coding and analysis to

full natural interactivity data annotation and analysis. The NITE objectives thus are: to develop a markup framework; identify, or develop, a number of natural interactivity best practice coding schemes to be described following the markup framework; and build a general-purpose natural interactivity annotation and analysis toolset which includes those coding schemes and supports the addition of new ones within the general boundaries of the markup framework. The NITE Project is funded by the European Commission to provide infrastructural technology for working with heavily cross-annotated multimodal data sets. This effort shares much in common with both the Annotation Graph Toolkit (Ma, Lee, Bird, & Maeda, 2002) and with ATLAS (Laprun, Fiscus, Garofolo, & Pajot, 2002). However, in keeping with the aim of supporting work with heavily cross-annotated data sets, NITE model allows easier access to rich structural information about the data than these other systems.

Although how annotations relate to time in the acoustic signal is important in corpus annotation, it has not been targeted specifically in our previous projects for the inherent difficulty of putting in direct relation abstract representations beyond word level with those at phone/word level like phonetic, phonological and prosodic representation. In particular phrase structures and sentences, are essentially structures built on top of other annotations (in our case, the words that make up an orthographic transcription) and have to derive their timings from the annotations on which they are based. Tree structures are common in describing a coherent sets of tags, but where several distinct types of annotation are present on the same material (syntax, discourse structure), the entire set may well not fit into a single tree. This is because different trees can draw on different leaves (discourse moves, words) and because even where they share the same leaves, they can draw on them in different and overlapping ways (e.g., disfluency and overlapping structure and syntax in relation to words).

As will be described in detail below, the problem of the annotation of overlaps has required a new coding of all the corpus AVIP/API in order to recover the temporal alignment of the phenomenon under study.

Our annotation activity has covered the items in the following list:

A. Elaboration and transformation of original texts

- normalization of texts containing dialogue transcription;
- transliteration of orthophonetic transcriptions in a standard orthographic format and creation of standard transliteration protocols;
- transformation of texts with overlaps organized on a dialogic basis (its content being assigned to the respective speaker), into texts with the overlaps temporally aligned with the corresponding acoustic signal;
- coding of the input file for the subsequent multilevel linguistic analysis in XML format adequate for its visualization in a standard commercial browser by means of href linking;
- creation of a file containing correspondences of all overlaps in XML format, between the original separate encoding of overlaps ascribed to each speaker in terms of turns and the transformed orthographic file where overlaps are encoded locally in each turn on a temporal basis;
- creation of a new archive of audio-files where each file contains the overlapped audio-materials coming from the two audio-tracks and collapsed in one single mono file. These files will then be linked to the new transcription of dialogues with temporally aligned overlappings.

B. Linguistic multilevel representation of each text at sentence level

- lexical annotation with association of lemmata to each wordform; association of a syntactic and a semantic class to each lemma;
- morphological annotation of each wordform with association of morphological features;
- syntactic annotation in bracketed constituents
- functional annotation in grammatical functions and transformation of the syntactic file
- containing wordforms of the orthographic text in a semantic representation into head lemmata and their features.
- anaphoric annotation of coreference between all referring expressions, both nominal and pronominal ones, without any restriction on

the type of reference as decided in the original MapTask, including both explicit and implicit linguistic elements.

2. Transforming the Orthophonetic file into the Orthographic file

As clarified above, the first step of our work has been that of going through the orthophonetically transcribed text in order to find regularities and then implement some set of regular expressions that could allow us to transform the nonlinguistic elements introduced in the audio transcription in some punctuation mark or else erase them from the text. The aim was that of preserving as much of the input transcribed text as possible in order to allow a direct link with it. We finally came up with the following list of transliteration rules:

- **# becomes '<' or '>'**
- **<eeh> and other interjections go without <>**
- **il<ll> una<aa> <aa>arco = erased material within <>**
- **<sp> (short pause) substituted by comma or dash.**
- **If at turn end can become period or ..., in that case only if the discourse is suspended.**
- **<eh!> becomes eh !**
- **/ indicates false start, substituted by comma.**
- **<eh?> becomes eh?**
- **des+ where + substituted by underscore**
- **<lp> (long pause) substituted by period, ... or - or ;**
- **<P> substituted by punctuation**

The above elements were all turned into some punctuation mark or else in case they indicated some fragment or interjection they were transformed in an orthographic corresponding value. We also listed all those symbols that constituted purely nonlinguistic semantically empty elements which were automatically erased:

<inspiration> , <laugh> , <vocal> , <breath>, <unclear>, <tongue-click>, <breathe> <NOISE>, <cough>, <clear-throath> [whispering], [dialect], {whispered}, [whispered]

The orthographic text was then used as input to the morphological analyser which produced a set of features associated to each wordform and a lemma. Words not belonging to standard Italian had to be listed in a user dictionary for an appropriate interpretation. We report here below the levels of representation introduced in our multilayer browser starting from the tokenized verticalized text.

3. From the Orthographic to the Temporaly Aligned File

Alla fine di questo lavoro di riorganizzazione del testo di partenza abbiamo prodotto due files in formato database, uno contenente tutti gli elementi linguistici e/o ortografici che appaiono nel file originale di trascrizione che chiamiamo "dgtdb04r_ort" e un secondo in cui sono stati aggiunti separatamente i segni di interpunzione che mancavano, e anche quelli che prima apparivano all'interno delle parentesi uncinate che chiamiano "dgtdb04r_punt_ort". Il secondo file ha subito anche la trasformazione relativa alla marcatura delle sovrapposizioni che hanno ricevuto un indice che ci permette di individuarle in maniera univoca nel testo. Mostriamo qui in basso la parte iniziale dei due files:

dgtdb04r_ort

```
p2#1:si    p2#1:
<inspiration>    no    nil
va'   si    va'
allora      si    allora
Giordano    si    Giordano
,     si    ,
<inspiration>    no    nil
senti si    senti
<sp>  ni    ,
<eeh> ni    eeh
allora      si    allora
il<ll>      ni    il
<ehm> ni    ehm
<inspiration>    no    nil
io    si    io
c'    si    c'
ho    si    ho
una   si    una
barca si    barca
<sp>  ni    ,
sul   si    sul
```

```
sul   si    sul
mare  si    mare
,     si    ,
#     no    nil
<p1#2>      no    nil
no    si    no
?     si    ?
#     si    #
p1#2: si    p1#2:
#     ni    ov_1
<p2#1>      no    nil
sullo si    sullo
sfondo      si    sfondo
?     si    ?
#     ni    >
p2#3: si    p2#3:
sì    si    sì
#     no    nil
<p1#4>      no    nil
<sp>  ni    -
<inspiration>     no    nil
<sp>  ni    -
<eeh> ni    eeh
#     si    #
p1#4: si    p1#4:
#     ni    ov_2
<p2#3>      no    nil
una   si    una
barca si    barca
a     si    a
vela  si    vela
#     ni    >
p2#5: si    p2#5:
<eh!> ni    eh
#     no    nil
<p1#6>      no    nil
<sp>  ni    -
c'    si    c'
è     si    è
una<aa>     ni    una
<ehm> ni    ehm
<sp>  ni    ,
#     si    #
una<aa>     ni    una
bandierina si     bandierina
```

dgtdb04r_punt_ort

```
p2#1: si    p2#1:
<inspiration>     no    nil
va'   si    va'
allora      si    allora
Giordano    si    Giordano
,     si    ,
<inspiration>     no    nil
senti si    senti
<sp>  ni    ,
<eeh> ni    eeh
allora      si    allora
il<ll>      ni    il
<ehm> ni    ehm
<inspiration>     no    nil
io    si    io
c'    si    c'
ho    si    ho
una   si    una
barca si    barca
<sp>  ni    ,
sul   si    sul
sul   si    sul
mare  si    mare
,     si    ,
#     no    nil
<p1#2>      no    nil
no    si    no
?     si    ?
#     si    #
p1#2: si    p1#2:
#     ni    ov_1
<p2#1>      no    nil
sullo si    sullo
sfondo      si    sfondo
?     si    ?
#     ni    >
p2#3: si    p2#3:
sì    si    sì
#     no    nil
<p1#4>      no    nil
<sp>  ni    -
<inspiration>     no    nil
<sp>  ni    -
<eeh> ni    eeh
#     si    #
      so    .
p1#4: si    p1#4:
#     ni    ov_2
<p2#3>      no    nil
una   si    una
barca si    barca
a     si    a
vela  si    vela
#     ni    >
p2#5: si    p2#5:
<eh!> ni    eh
```

```
#       no      nil
<p1#6>          no      nil
<sp>    ni      -
c'      si      c'
è       si      è
una<aa>         ni      una
<ehm>   ni      ehm
<sp>    ni      ,
#       si      #
una<aa>         ni      una
bandierina      si      bandierina
        so      .
```

Come si può facilmente notare i due spezzoni di files sono di lunghezza diversa: il primo, l'originale contiene solo ed esclusivamente gli elementi linguistici e ortografici del file originale ed è di 66 elementi; il secondo invece, a cui è stata aggiunta la punteggiatura mancante è di 68 elementi. In un lavoro successivo abbiamo quindi proceduto al riallineamento dei turni su base temporale, spostando il materiale di sovrapposizione dal turno in cui si trovava nel file iniziale alla sua posizione effettiva, nel luogo di sovrapposizione. Abbiamo poi marcato di rosso le parti del file originale che sono state modificate sulla base dell'allineamento temporale. Questo nuovo file lo abbiamo chiamato "dgtdb04r_sovr_ort":

"dgtdb04r_sovr_ort"

```
p2#1: si      p2#1:
<inspiration>           no      nil
va'     si      va'
allora          si      allora
Giordano        si      Giordano
,       si      ,
<inspiration>           no      nil
senti   si      senti
<sp>    ni      ,
<eeh>   ni      eeh
allora          si      allora
il<ll>          ni      il
<ehm>   ni      ehm
<inspiration>           no      nil
io      si      io
c'      si      c'
ho      si      ho
una     si      una
barca   si      barca
<sp>    ni      ,
sul     si      sul
sul     si      sul
mare    si      mare
,       si      ,
#       ni      ov_1
<p2#1>          no      nil
sullo si        sullo
sfondo          si      sfondo
?       si      ?
#       ni      >
#       no      nil
<p1#2>          no      nil
no      si      no
?       si      ?
#       si      #
p1#2: si      p1#2:
p2#3: si      p2#3:
sì      si      sì
#       ni      ov_2
<p2#3>          no      nil
una     si      una
barca   si      barca
a       si      a
vela    si      vela
#       ni      >
#       no      nil
<p1#4>          no      nil
<sp>    ni      -
<inspiration>           no      nil
<sp>    ni      -
<eeh>   ni      eeh
#       si      #
        so      .
p1#4: si      p1#4:
p2#5: si      p2#5:
<eh!>   ni      eh
#       ni      ov_3
<p2#5>          no      nil
sì      si      sì
<sp>    ni      .
#       ni      >
#       no      nil
<p1#6>          no      nil
<sp>    ni      -
c'      si      c'
è       si      è
una<aa>         ni      una
<ehm>   ni      ehm
<sp>    ni      ,
#       si      #
una<aa>         ni      una
bandierina      si      bandierina
        so      .
```

Come si può notare, dalla operazione di riallineamento le sovrapposizioni che fisicamente si realizzano nello spazio testuale che termina con la parola "bandierina" sono diventate 3. Alcuni turni risultano essere vuoti dal momento che il loro contenuto linguistico viene realizzato come sovrapposizione di altro materiale

linguistico. In questi files iniziali sono anche contenuti i marcatori di fenomeni paralinguistici che verranno eliminati dall'analisi successiva e che riportano nella colonna centrale la marca "no".
Da questi files viene quindi prodotto un nuovo file di testo verticalizzato in cui sono stati espunti tutti gli elementi marcati con "no". Il testo da cui parte il lavoro di analisi linguistica è quindi costituito da questi files iniziali dei quali viene prodotta una versione in formato xml, di cui di nuovo riportiamo le parti relative alla porzione di testo mostrata qui sopra.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ortofonetic_file>
<turn id="p2#1">
<w id=" ortfon_1 "type=" si ">  p2#1: </w>
<w id=" ortfon_2 "type=" no ">  <inspiration>
</w>
<w id=" ortfon_3 "type=" si ">  va' </w>
<w id=" ortfon_4 "type=" si ">  allora </w>
<w id=" ortfon_5 "type=" si ">  Giordano </w>
<w id=" ortfon_6 "type=" si ">  , </w>
<w id=" ortfon_7 "type=" no ">  <inspiration>
</w>
<w id=" ortfon_8 "type=" si ">  senti </w>
<w id=" ortfon_9 "type=" si ">  <sp> </w>
<w id=" ortfon_10 "type=" si ">  <eeh> </w>
<w id=" ortfon_11 "type=" si ">  allora </w>
<w id=" ortfon_12 "type=" si ">  il<ll> </w>
<w id=" ortfon_13 "type=" si ">  <ehm> </w>
<w id=" ortfon_14 "type=" no ">
<inspiration> </w>
<w id=" ortfon_15 "type=" si ">  io </w>
<w id=" ortfon_16 "type=" si ">  c' </w>
<w id=" ortfon_17 "type=" si ">  ho </w>
<w id=" ortfon_18 "type=" si ">  una </w>
<w id=" ortfon_19 "type=" si ">  barca </w>
<w id=" ortfon_20 "type=" si ">  <sp> </w>
<w id=" ortfon_21 "type=" si ">  sul </w>
<w id=" ortfon_22 "type=" si ">  sul </w>
<w id=" ortfon_23 "type=" si ">  mare </w>
<w id=" ortfon_24 "type=" si ">  , </w>
<w id=" ortfon_25 "type=" si ">  # </w>
<w id=" ortfon_26 "type=" no ">  <p2#1> </w>
<w id=" ortfon_27 "type=" si ">  sullo </w>
<w id=" ortfon_28 "type=" si ">  sfondo </w>
<w id=" ortfon_29 "type=" si ">  ? </w>
<w id=" ortfon_30 "type=" si ">  # </w>
<w id=" ortfon_31 "type=" no ">  # </w>
<w id=" ortfon_32 "type=" no ">  <p1#2> </w>
<w id=" ortfon_33 "type=" si ">  no </w>
<w id=" ortfon_34 "type=" si ">  ? </w>
<w id=" ortfon_35 "type=" si ">  # </w>
</turn>
```

Questo primo file conserva la memoria del file ortografico originale con gli indici che registrano il numero di tokens iniziali, corrispondente a 4761. Il file ortofonetico è corrispondente a quello denominato "dgtdb04r_ort" più sopra. Invece il nuovo file ortografico, corrispondente al file denominato "dgtdb04r_punt_ort" ripulito però dagli elementi ridondanti, e che riportiamo in basso in formato xml, ha la nuova riorganizzato dei turni e della punteggiatura e dei nuovi indici con gli href che puntano a quello riportato qui sopra, che noi abbiamo chiamato file ortofonetico. Come è possibile notare, in alcuni casi alcuni elementi del file ortofonetico vengono saltati, sono cioè privi di href: i due files non sono isomorfi e il mapping viene fatto dall'alto in basso, ma non è possibile il contrario. Insomma il nuovo file su cui viene prodotta l'analisi sintattica è un sottoinsieme proprio di quello di partenza, difatti il numero complessivo di tokens del nuovo files è di 4286.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ortosovr_word_file>
<turn id="p2#1">
<w id=" ort_1 " href=" dgtdb04r #id( ortfon_1
)">  p2#1: </w>
<w id=" ort_2 " href=" dgtdb04r #id( ortfon_3
)">  va' </w>
<w id=" ort_3 " href=" dgtdb04r #id( ortfon_4
)">  allora </w>
<w id=" ort_4 " href=" dgtdb04r #id( ortfon_5
)">  Giordano </w>
<w id=" ort_5 " href=" dgtdb04r #id( ortfon_6
)">  , </w>
<w id=" ort_6 " href=" dgtdb04r #id( ortfon_8
)">  senti </w>
<w id=" ort_7 " href=" dgtdb04r #id( ortfon_9
)">  , </w>
<w id=" ort_8 " href=" dgtdb04r #id(
ortfon_10 )">  eeh </w>
<w id=" ort_9 " href=" dgtdb04r #id(
ortfon_11 )">  allora </w>
<w id=" ort_10 " href=" dgtdb04r #id(
ortfon_12 )">  il </w>
<w id=" ort_11 " href=" dgtdb04r #id(
ortfon_13 )">  ehm </w>
<w id=" ort_12 " href=" dgtdb04r #id(
ortfon_15 )">  io </w>
<w id=" ort_13 " href=" dgtdb04r #id(
ortfon_16 )">  c' </w>
<w id=" ort_14 " href=" dgtdb04r #id(
ortfon_17 )">  ho </w>
<w id=" ort_15 " href=" dgtdb04r #id(
ortfon_18 )">  una </w>
<w id=" ort_16 " href=" dgtdb04r #id(
ortfon_19 )">  barca </w>
<w id=" ort_17 " href=" dgtdb04r #id(
ortfon_20 )">  , </w>
<w id=" ort_18 " href=" dgtdb04r #id(
ortfon_21 )">  sul </w>
<w id=" ort_19 " href=" dgtdb04r #id(
ortfon_22 )">  sul </w>
<w id=" ort_20 " href=" dgtdb04r #id(
ortfon_23 )">  mare </w>
<w id=" ort_21 " href=" dgtdb04r #id(
ortfon_24 )">  , </w>
<w id=" ort_22 " href=" dgtdb04r #id(
ortfon_25 )">  ov_1 </w>
<w id=" ort_23 " href=" dgtdb04r #id(
ortfon_27 )">  sullo </w>
<w id=" ort_24 " href=" dgtdb04r #id(
ortfon_28 )">  sfondo </w>
```

```
<w id=" ort_25 " href=" dgtdb04r #id(
ortfon_29 )">  ? </w>
<w id=" ort_26 " href=" dgtdb04r #id(
ortfon_30 )">  > </w>
<w id=" ort_27 " href=" dgtdb04r #id(
ortfon_33 )">  no </w>
<w id=" ort_28 " href=" dgtdb04r #id(
ortfon_34 )">  ? </w>
<w id=" ort_29 " href=" dgtdb04r #id(
ortfon_35 )">  # </w>
</turn>
```

Non riportiamo qui anche tutti i files intermedi che sono serviti ai programmi in PROLOG per produrre l'output xml che abbiamo mostrato qui sopra. La versione del file finale xml riportata qui sopra è stata poi ulteriormente arricchita dell'informazione temporale attaccata ad ogni turno

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ortoword_file>
<time id="00:04.236 - 00:12.938">
<turn id="p2#1">
<w id=" ort_1 " href=" dgtdb04r #id( ortfon_1
)">  p2#1: </w>
<w id=" ort_2 " href=" dgtdb04r #id( ortfon_3
)">  va' </w>
<w id=" ort_3 " href=" dgtdb04r #id( ortfon_4
)">  allora </w>
<w id=" ort_4 " href=" dgtdb04r #id( ortfon_5
)">  Giordano </w>
<w id=" ort_5 " href=" dgtdb04r #id( ortfon_6
)">  , </w>
<w id=" ort_6 " href=" dgtdb04r #id( ortfon_8
)">  senti </w>
<w id=" ort_7 " href=" dgtdb04r #id( ortfon_9
)">  , </w>
<w id=" ort_8 " href=" dgtdb04r #id(
ortfon_10 )">  eeh </w>
<w id=" ort_9 " href=" dgtdb04r #id(
ortfon_11 )">  allora </w>
<w id=" ort_10 " href=" dgtdb04r #id(
ortfon_12 )">  il </w>
<w id=" ort_11 " href=" dgtdb04r #id(
ortfon_13 )">  ehm </w>
<w id=" ort_12 " href=" dgtdb04r #id(
ortfon_15 )">  io </w>
<w id=" ort_13 " href=" dgtdb04r #id(
ortfon_16 )">  c' </w>
<w id=" ort_14 " href=" dgtdb04r #id(
ortfon_17 )">  ho </w>
<w id=" ort_15 " href=" dgtdb04r #id(
ortfon_18 )">  una </w>
<w id=" ort_16 " href=" dgtdb04r #id(
ortfon_19 )">  barca </w>
<w id=" ort_17 " href=" dgtdb04r #id(
ortfon_20 )">  , </w>
<w id=" ort_18 " href=" dgtdb04r #id(
ortfon_21 )">  sul </w>
<w id=" ort_19 " href=" dgtdb04r #id(
ortfon_22 )">  sul </w>
<w id=" ort_20 " href=" dgtdb04r #id(
ortfon_23 )">  mare </w>
<w id=" ort_21 " href=" dgtdb04r #id(
ortfon_24 )">  , </w>
<w id=" ort_22 " href=" dgtdb04r #id(
ortfon_27 )">  no </w>
<w id=" ort_23 " href=" dgtdb04r #id(
ortfon_28 )">  ? </w>
<w id=" ort_24 " href=" dgtdb04r #id(
ortfon_29 )">  # </w>
</turn>
</time>
```

Il prodotto finale di questo lavoro di riorganizzazione e riclassificazione della trascrizione ortografica di partenza viene utilizzato per visualizzare il testo sotto browser – nel nostro sito all'indirizzo seguente:
http://sisley.cgm.unive.it/HTMLipar/index.htm
Nel sito è possibile isolare le sovrapposizioni che sono così direttamente udibili dal file wav corrispondente che noi abbiamo creato ritagliando ed unendo le parti di segnale acustico che nelle due tracce risulta essere coincidente temporalmente. Le sovrapposizioni a cui si accede dal file ortofonetico originale, sono anche leggibili nella loro interezza visto che abbiamo prodotto da programma anche tutti gli spezzoni di testo che le costituiscono. Riportiamo qui in basso gli spezzoni di testo relativi alla prima sovrapposizione:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<sovrword_file>
<overlap  id="ov_1">
<w id=" ortsovr_25 " ortofon_tok=" # " href="
dgtdb04r_ortofonfile #id( ortfon_25 )">  ov_1
</w>
<w id=" ortsovr_26 " ortofon_tok=" p1#2 "
href=" dgtdb04r_ortofonfile #id( ortfon_26
)">  nil </w>
<w id=" ortsovr_27 " ortofon_tok=" no "
href=" dgtdb04r_ortofonfile #id( ortfon_27
)">  sullo </w>
<w id=" ortsovr_28 " ortofon_tok=" ? " href="
dgtdb04r_ortofonfile #id( ortfon_28 )">
sfondo </w>
<w id=" ortsovr_29 " ortofon_tok=" # " href="
dgtdb04r_ortofonfile #id( ortfon_29 )">  ?
</w>
<w id=" ortsovr_30 " ortofon_tok=" p1#2: "
href=" dgtdb04r_ortofonfile #id( ortfon_30
)">  > </w>
<w id=" ortsovr_31 " ortofon_tok=" # " href="
dgtdb04r_ortofonfile #id( ortfon_31 )">  nil
</w>
<w id=" ortsovr_32 " ortofon_tok=" p2#1 "
href=" dgtdb04r_ortofonfile #id( ortfon_32
)">  nil </w>
<w id=" ortsovr_33 " ortofon_tok=" sullo "
href=" dgtdb04r_ortofonfile #id( ortfon_33
)">  no </w>
<w id=" ortsovr_34 " ortofon_tok=" sfondo "
href=" dgtdb04r_ortofonfile #id( ortfon_34
)">  ? </w>
<w id=" ortsovr_35 " ortofon_tok=" ? " href="
dgtdb04r_ortofonfile #id( ortfon_35 )">  #
</w>
<w id=" ortsovr_36 " ortofon_tok=" # " href="
dgtdb04r_ortofonfile #id( ortfon_36 )">
p1#2: </w>
```

```
</overlap>
```

Ogni “overlap” ha il proprio href collegato al file ortofonetico che come si può ricavare guardando il file corrispondente riportato più sopra coincidono con gli indici che vanno da ortfon_25 a 35.

4. Tokenizzazione e Analisi sintattica

Lo scopo della creazione del database era quello di avere uno strumento che ci fornisse con maggiore accuratezza e consistenza il file su cui intervenire con i programmi di analisi semiautomatica, cioè quello per la tokenizzazione, il tagging automatico, per l'analisi lessicale e morfologica del testo taggato e quello per l'analisi sintattica.
Abbiamo quindi proceduto alla tokenizzazione e al tagging automatico del testo comune sul quale abbiamo compiuto la disambiguazione semi-automatica di tutti i 4282 tokens costituenti il file.
A questo punto abbiamo lanciato il parser che per aggiustamenti successivi ha compiuto una prima analisi in costituenti sintattici, analisi che è stata poi verificata manualmente attraverso gli strumenti di validazione e verifica disponibili a Venezia e con il lavoro di un esperto in annotazione sintattica.

4.1 Tagging – prima parte

Il lavoro di tagging viene svolto in quattro fasi: la tokenizzazione, l’analisi morfologica e lessicale, la disambiguazione, la lemmatizzazione. La tokenizzazione include anche la produzione di forme polirematiche o di locuzioni che da elementi linguistici semplici diventano complessi. Nel nostro testo sono state trovate le seguenti forme:
barca_a_vela
in_giù
all_insù
fuori_dalla
come_se
sopra_all
sotto_al
accanto_all
una_specie_di
che_cosa
più_o_meno
in_mezzo
in_tutto
rispetto_alla
giù_di_lì
per_niente
nel_senso_che
Allo stesso tempo le parole amalgamate come i verbi cliticizzati vengono decomposti e appaiono in due tokens separati. Una volta ottenuto il file di testo taggato questo viene passato attraverso il programma di disambiguazione che in modalità interattiva permette di indicare direttamente l’etichetta linguistica più adeguata a un particolare contesto anche in contrasto con quanto previsto dal programma stesso. Il file disambiguata riporta una serie di informazioni relative alle etichette sintattiche di riferimento per la scelta operata, nonché un peso che misura il grado di affidabilità della scelta stessa. Riportiamo di nuovo la stessa porzione di testo con le etichette disambiguate:

```
i(1-0, [cp]-p2_1-cp, [turn]-turn-0, nil).
i(2-0, [svt]-và-cp, [intj]-intj-1000, 0).
i(3-0, [fc, sa]-allora-cp, [avv, congf, agn]-
congf-100, 42).
i(4-0, [sn]-'Giordano'-sn, [nh]-nh-10, 192).
i(5-0, [fp]-',-fp, [punt]-punt-1000, nil).
i(6-0, [ibar, ir_infl]-senti-ibar, [vin,
virin, virt, vt]-vin-1000, 255).
i(7-0, [fp]-',-fp, [punt]-punt-1000, nil).
i(8-0, [svt]-eeh-cp, [intj]-intj-1000, 1267).
i(9-0, [fc, sa]-allora-cp, [avv, congf, agn]-
congf-100, 42).
i(10-0, [sn]-il-sn, [art]-art-10, 1314).
i(11-0, [svt]-ehm-cp, [intj]-intj-1000,
1380).
i(12-0, [sn]-io-sn, [pron]-pron-10, 1427).
i(13-0, [ibar]-c-ibar, [clit, clitabl,
clitdat, pron]-expl-1000, 81259).
i(14-0, [ibar]-ho-ibar, [ausa, vc]-vc-1,
1577).
i(15-0, [sn]-una-sn, [num, art]-art-10,
1723).
i(16-0, [sn]-barca-sn, [n]-n-1, 1768).
i(17-0, [fp]-',-fp, [punt]-punt-1, nil).
i(18-0, [sp]-sul-sp, [part]-part-10, 1827).
i(19-0, [sp]-sul-sp, [part]-part-10, 1827).
i(20-0, [sn]-mare-sn, [n]-n-10, 1916).
i(21-0, [fp]-',-fp, [punt]-punt-1, nil).
i(22-0, [fp]-ov_1-fp, [overlap]-overlap-1000,
nil).
i(23-0, [sp]-sullo-sp, [part]-part-10, 1973).
i(24-0, [ibar, sn]-sfondo-sn, [n, vin, vt]-n-
10, 2064).
i(25-0, [cp]- ? -cp, [puntint]-puntint-1000,
nil).
i(26-1, [fp]-(>)-fp, [par]-par-1000, nil).
```

```
i(27-1, [svt]-no-cp, [intj, n]-intj-1000,
2316).
i(28-1, [cp]- ? -cp, [puntint]-puntint-1000,
nil).
i(29-2, [fp]- # -cp, [overlap]-overlap-1000,
nil).
i(30-2, [cp]-p1_2-cp, [turn]-turn-1000, nil).
i(31-2, [cp]-p2_3-cp, [turn]-turn-1, nil).
i(32-2, [svt]-sì-cp, [in, intj]-intj-1000,
2391).
i(33-2, [fp]-ov_2-cp, [overlap]-overlap-1000,
nil).
i(34-2, [sn]-una-sn, [num, art]-art-10,
1723).
i(35-2, [sn]-barca_a_vela-sn, [n]-n-1, 2457).
i(36-2, [fp]-(>)-fp, [par]-par-1000, nil).
i(37-2, [fp]-(-)-fp, [par]-par-1, nil).
i(38-2, [fp]-(-)-fp, [par]-par-1, nil).
i(39-2, [svt]-eeh-cp, [intj]-intj-1000,
1267).
i(40-2, [fp]- # -cp, [overlap]-overlap-1000,
nil).
i(41-2, [cp]-'''.'''-cp, [punto]-punto-1000,
nil).
i(42-3, [cp]-p1_4-cp, [turn]-turn-1, nil).
i(43-3, [cp]-p2_5-cp, [turn]-turn-1, nil).
i(44-3, [svt]-eh-cp, [intj]-intj-1000, 2537).
i(45-3, [fp]-ov_3-cp, [overlap]-overlap-1000,
nil).
i(46-3, [svt]-sì-cp, [in, intj]-intj-1000,
2391).
i(47-3, [fp]-(>)-fp, [par]-par-1000, nil).
i(48-3, [fp]-(-)-fp, [par]-par-1, nil).
i(49-3, [ibar]-c-ibar, [clit, clitabl,
clitdat, pron]-clitabl-1000, 1488).
i(50-3, [ibar]-è-ibar, [ause, vc]-vc-1,
2583).
i(51-3, [sn]-una-sn, [num, art]-art-10,
1723).
i(52-3, [svt]-ehm-cp, [intj]-intj-1000,
1380).
i(53-3, [fp]-(-)-fp, [par]-par-1000, nil).
i(54-3, [fp]- # -fp, [overlap]-overlap-1000,
nil).
i(55-3, [sn]-una-sn, [num, art]-art-10,
1723).
i(56-3, [sn]-bandierina-sn, [n]-n-1, 2730).
i(57-3, [cp]-'''.'''-cp, [punto]-punto-1000,
nil).
```

Una volta costruito e controllato il file taggato e disambiguato si può procedere alla fase di lemmatizzazione. Per questo scopo è necessario utilizzare un altri file prodotto in fase di tagging, cioè il file di features, contenente tutti i tipi prodotti dall'analisi dei tokens e riportati una volta sola, con associate a ciascuna parola tutte le informazioni lessicali, morfologiche, sintattiche e semantiche per ciascuna possibile interpretazione registrata dal programma sulla base delle proprie conoscenze. Il file di feats per le prime 57 parole del testo è il seguente:

```
0-sw(1-và-[intj]-1-[intj-và-[cat=intj]]).
42-sw(2-allora-[avv, congf, agn]-3-[agn-
allora-[cat=adj_noun, type=abst, gen=m], avv-
allora-[type=z], avv-allora-[type=t], congf-
allora-[type=sum]]).
192-sw(3-'Giordano'-[nh]-1-[nh-'Giordano'-
[type=hum, gen=m]]).
255-sw(4-senti-[vin, virin, virt, vt]-4-[vin-
sent-[mood=indic, tense=pres, pers=2, num=s,
scat=intr], virin-sent-[mood=subj,
tense=pres, pers=1, num=s, scat=intr], virin-
sent-[mood=subj, tense=pres, pers=2, num=s,
scat=intr], virin-sent-[mood=subj,
tense=pres, pers=3, num=s, scat=intr], virin-
sent-[mood=imp, tense=pres, pers=3, num=s,
scat=intr], vt-sent-[mood=indic, tense=pres,
pers=2, num=s, scat=intr], virt-sent-
[mood=subj, tense=pres, pers=1, num=s,
scat=intr], virt-sent-[mood=subj, tense=pres,
pers=2, num=s, scat=intr], virt-sent-
[mood=subj, tense=pres, pers=3, num=s,
scat=intr], virt-sent-[mood=imp, tense=pres,
pers=3, num=s, scat=intr], vin-sent-
[mood=indic, tense=pres, pers=2, num=s,
scat=intr], virin-sent-[mood=imp, tense=pres,
pers=2, num=s, scat=intr], vt-sent-
[mood=indic, tense=pres, pers=2, num=s,
scat=rifl], virt-sent-[mood=imp, tense=pres,
pers=2, num=s, scat=rifl], vt-sent-
[mood=indic, tense=pres, pers=2, num=s,
scat=tr], virt-sent-[mood=imp, tense=pres,
pers=2, num=s, scat=tr]]).
1267-sw(5-eeh-[intj]-1-[intj-eeh-
[cat=intj]]).
1314-sw(6-il-[art]-1-[art-il-[type=def,
pred=il, gen=m, num=s]]).
1380-sw(7-ehm-[intj]-1-[intj-ehm-
[cat=intj]]).
1427-sw(8-io-[pron]-1-[pron-io-[type=pers,
gen=fm, num=s]]).
1488-sw(9-c-[clit, clitabl, clitdat, pron]-4-
[ci, ci-cio-ci, ci-cio-ci, ci, ci-cio-ci]).
1577-sw(10-ho-[ausa, vc]-2-[ausa-av-
[mood=indic, tense=pres, pers=1, num=s,
scat=aux], vc-av-[mood=indic, tense=pres,
pers=1, num=s, scat=cop]]).
1723-sw(11-una-[num, art]-2-[un-un, un-un]).
1768-sw(12-barca-[n]-1-[n-barc-[type=com,
gen=f, num=s]]).
1827-sw(13-sul-[part]-1-[part-su-[cat1=prep,
p2=il, cat2=art, type=det, gen=m, num=s]]).
1916-sw(14-mare-[n]-1-[n-mar-[type=com,
gen=m, num=s]]).
1973-sw(15-sullo-[part]-1-[part-su-
[cat1=prep, p2=il, cat2=art, type=det, gen=m,
num=s]]).
2064-sw(16-sfondo-[n, vin, vt]-3-[n-sfond-
[type=com, gen=m, num=s], vin-sfond-
[mood=indic, tense=pres, pers=1, num=s,
scat=intr], vt-sfond-[mood=indic, tense=pres,
pers=1, num=s, scat=intr], vt-sfond-
[mood=indic, tense=pres, pers=1, num=s,
scat=tr]]).
2316-sw(17-no-[intj, n]-2-[intj-no-
[cat=intj], n-no-[type=invar, gen=m]]).
2391-sw(18-sì-[in, intj]-2-[in-sì-[type=q],
intj-sì-[cat=intj]]).
2457-sw(19-barca_a_vela-[n]-1-[n-[barca, a,
vela]-[type=invar, gen=f, num=s]]).
2537-sw(20-eh-[intj]-1-[intj-eh-[cat=intj]]).
2583-sw(21-è-[ause, vc]-2-[ause-ess-
[mood=indic, tense=pres, pers=3, num=s,
scat=aux], vc-ess-[mood=indic, tense=pres,
pers=3, num=s, scat=cop]]).
2730-sw(22-bandierina-[n]-1-[n-bandierin-
[type=com, gen=f, num=s]]).
```

Questo file verrà utilizzato dal programma di lemmatizzazione per assegnare i tratti e il lemma deciso dalla disambiguazione categoriale. Nel caso in cui ci siano ancora ambiguità – ad esempio, la stessa categoria lessicale ma diversi lemma (state) – è necessario intervenire a mano successivamente. Il file lemmatizzato verrà poi utilizzato per produrre il file mfeats in formato xml. Mostriamo di nuovo la porzione di testo in formato lemmatizzazione:

```
i(p2_1, turn, p2_1).
i(và, intj, và-[cat=intj]).
i(allora, congf, allora-[type=sum]).
i('Giordano', nh, 'Giordano'-[feat=hum]).
i(',', punt, ',').
i(senti, vin, sentire-[mood=indic,
tense=pres, pers=2, num=s, scat=intr]).
i(',', punt, ',').
i(eeh, intj, eeh-[cat=intj]).
i(allora, congf, allora-[type=sum]).
i(il, art, il-[type=def, pred=il, gen=m,
num=s]).
i(ehm, intj, ehm-[cat=intj]).
i(io, pron, io-[type=pers, gen=fm, num=s]).
i(c, expl, ci).
i(ho, vc, avere-[mood=indic, tense=pres,
pers=1, num=s, scat=cop]).
i(una, art, un-[type=ind, pred=un, gen=f,
num=s]).
i(barca, n, barca-[type=com, gen=f, num=s]).
i(',', punt, ',').
i(sul, part, su-[cat1=prep, p2=il, cat2=art,
type=det, gen=m, num=s]).
i(sul, part, su-[cat1=prep, p2=il, cat2=art,
type=det, gen=m, num=s]).
i(mare, n, mare-[type=com, gen=m, num=s]).
i(',', punt, ',').
i(ov_1, overlap, ov_1).
i(sullo, part, su-[cat1=prep, p2=il,
cat2=art, type=det, gen=m, num=s]).
i(sfondo, n, sfondo-[type=com, gen=m,
num=s]).
i(?, puntint, ?).
i(>, par, >).
i(no, intj, no-[cat=intj]).
i(?, puntint, ?).
i(#, overlap, #).
i(p1_2, turn, p1_2).
i(p2_3, turn, p2_3).
i(sì, intj, sì-[cat=intj]).
i(ov_2, overlap, ov_2).
i(una, art, un-[type=ind, pred=un, gen=f,
num=s]).
i(barca_a_vela, n, [barca,a,vela]-
[type=invar, gen=f, num=s]).
i(>, par, >).
i(-, par, -).
i(-, par, -).
i(eeh, intj, eeh-[cat=intj]).
i(#, overlap, #).
i('''.''', punto, '''.''').
i(p1_4, turn, p1_4).
i(p2_5, turn, p2_5).
i(eh, intj, eh-[cat=intj]).
i(ov_3, overlap, ov_3).
i(sì, intj, sì-[cat=intj]).
i(>, par, >).
i(-, par, -).
i(c, clitabl, ci-[case=abl, pers=1, num=p,
gen=mf]).
i(è, vc, essere-[mood=indic, tense=pres,
pers=3, num=s, scat=cop]).
i(una, art, un-[type=ind, pred=un, gen=f,
num=s]).
i(ehm, intj, ehm-[cat=intj]).
i(-, par, -).
i(#, overlap, #).
i(una, art, un-[type=ind, pred=un, gen=f,
num=s]).
i(bandierina, n, bandierina-[type=com, gen=f,
num=s]).
i('''.''', punto, '''.''').
```

Per finire, mostriamo il file mfeats in formato xml che viene prodotto dal programma sulla base di quello lemmatizzato opportunamente verificato e corretto manualmente:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mword_file id="dgtdb04r">
<mw id="mw_0" pos="I" sfeats="turn"
href="toks#id(w_0)"> p2_1</mw>
<mw id="mw_1" pos="I" mfeats="NN" lemma="và"
sfeats="intj" href="toks#id(w_1)"> và</mw>
<mw id="mw_2" pos="C" mfeats="NN"
lemma="allora" sfeats="congf" sems="sum"
href="toks#id(w_2)"> allora</mw>
<mw id="mw_3" pos="N" mfeats="ms"
lemma="Giordano" sfeats="nh" sems="hum"
href="toks#id(w_3)"> Giordano</mw>
<mw id="mw_4" pos="PU" sfeats="punt"
href="toks#id(w_4)"> ,</mw>
<mw id="mw_5" pos="V" mfeats="KL2s"
lemma="sentire" sfeats="vin" sems="intr"
href="toks#id(w_5)"> senti</mw>
<mw id="mw_6" pos="PU" sfeats="punt"
href="toks#id(w_6)"> ,</mw>
<mw id="mw_7" pos="I" mfeats="NN" lemma="eeh"
sfeats="intj" href="toks#id(w_7)"> eeh</mw>
<mw id="mw_8" pos="C" mfeats="NN"
lemma="allora" sfeats="congf" sems="sum"
href="toks#id(w_8)"> allora</mw>
<mw id="mw_9" pos="D" mfeats="ms" lemma="il"
sfeats="art" sems="def" href="toks#id(w_9)">
il</mw>
<mw id="mw_10" pos="I" mfeats="NN"
lemma="ehm" sfeats="intj"
href="toks#id(w_10)"> ehm</mw>
<mw id="mw_11" pos="E" mfeats="fms"
lemma="io" sfeats="pron" sems="pers"
href="toks#id(w_11)"> io</mw>
<mw id="mw_12" pos="E" sfeats="expl"
href="toks#id(w_12)"> c</mw>
<mw id="mw_13" pos="V" mfeats="KL1s"
lemma="avere" sfeats="vc" sems="cop"
href="toks#id(w_13)"> ho</mw>
<mw id="mw_14" pos="D" mfeats="fs" lemma="un"
sfeats="art" sems="ind" href="toks#id(w_14)">
una</mw>
<mw id="mw_15" pos="N" mfeats="fs"
lemma="barca" sfeats="n" sems="com"
href="toks#id(w_15)"> barca</mw>
<mw id="mw_16" pos="PU" sfeats="punt"
href="toks#id(w_16)"> ,</mw>
<mw id="mw_17" pos="P" mfeats="ms" lemma="su"
sfeats="part" sems="def"
href="toks#id(w_17)"> sul</mw>
```

```
<mw id="mw_18" pos="P" mfeats="ms" lemma="su"
sfeats="part" sems="def"
href="toks#id(w_18)"> sul</mw>
<mw id="mw_19" pos="N" mfeats="ms"
lemma="mare" sfeats="n" sems="com"
href="toks#id(w_19)"> mare</mw>
<mw id="mw_20" pos="PU" sfeats="punt"
href="toks#id(w_20)"> ,</mw>
<mw id="mw_21" pos="I" sfeats="overlap"
href="toks#id(w_21)"> ov_1</mw>
<mw id="mw_22" pos="P" mfeats="ms" lemma="su"
sfeats="part" sems="def"
href="toks#id(w_22)"> sullo</mw>
<mw id="mw_23" pos="N" mfeats="ms"
lemma="sfondo" sfeats="n" sems="com"
href="toks#id(w_23)"> sfondo</mw>
<mw id="mw_24" pos="PU" sfeats="puntint"
href="toks#id(w_24)"> ?</mw>
<mw id="mw_25" pos="PU" sfeats="par"
href="toks#id(w_25)"> ></mw>
<mw id="mw_26" pos="I" mfeats="NN" lemma="no"
sfeats="intj" href="toks#id(w_26)"> no</mw>
<mw id="mw_27" pos="PU" sfeats="puntint"
href="toks#id(w_27)"> ?</mw>
<mw id="mw_28" pos="PU" sfeats="overlap"
href="toks#id(w_28)"> #</mw>
<mw id="mw_29" pos="I" sfeats="turn"
href="toks#id(w_29)"> p1_2</mw>
<mw id="mw_30" pos="I" sfeats="turn"
href="toks#id(w_30)"> p2_3</mw>
<mw id="mw_31" pos="I" mfeats="NN" lemma="sì"
sfeats="intj" href="toks#id(w_31)"> sì</mw>
<mw id="mw_32" pos="I" sfeats="overlap"
href="toks#id(w_32)"> ov_2</mw>
<mw id="mw_33" pos="D" mfeats="fs" lemma="un"
sfeats="art" sems="ind" href="toks#id(w_33)">
una</mw>
<mw id="mw_34" pos="N" mfeats="fs"
lemma="barca_a_vela" sfeats="n" sems="invar"
href="toks#id(w_34)..id(w_35)..id(w_36)">
barca a vela</mw>
<mw id="mw_35" pos="PU" sfeats="par"
href="toks#id(w_37)"> ></mw>
<mw id="mw_36" pos="PU" sfeats="par"
href="toks#id(w_38)"> -</mw>
<mw id="mw_37" pos="PU" sfeats="par"
href="toks#id(w_39)"> -</mw>
<mw id="mw_38" pos="I" mfeats="NN"
lemma="eeh" sfeats="intj"
href="toks#id(w_40)"> eeh</mw>
<mw id="mw_39" pos="PU" sfeats="overlap"
href="toks#id(w_41)"> #</mw>
<mw id="mw_40" pos="PU" sfeats="punto"
href="toks#id(w_42)"> .</mw>
<mw id="mw_41" pos="I" sfeats="turn"
href="toks#id(w_43)"> p1_4</mw>
<mw id="mw_42" pos="I" sfeats="turn"
href="toks#id(w_44)"> p2_5</mw>
<mw id="mw_43" pos="I" mfeats="NN" lemma="eh"
sfeats="intj" href="toks#id(w_45)"> eh</mw>
<mw id="mw_44" pos="I" sfeats="overlap"
href="toks#id(w_46)"> ov_3</mw>
<mw id="mw_45" pos="I" mfeats="NN" lemma="sì"
sfeats="intj" href="toks#id(w_47)"> sì</mw>
<mw id="mw_46" pos="PU" sfeats="par"
href="toks#id(w_48)"> ></mw>
<mw id="mw_47" pos="PU" sfeats="par"
href="toks#id(w_49)"> -</mw>
<mw id="mw_48" pos="E" mfeats="1mfp"
lemma="ci" sfeats="clitabl" sems="abl"
href="toks#id(w_50)"> c</mw>
<mw id="mw_49" pos="V" mfeats="KL3s"
lemma="essere" sfeats="vc" sems="cop"
href="toks#id(w_51)"> è</mw>
<mw id="mw_50" pos="D" mfeats="fs" lemma="un"
sfeats="art" sems="ind" href="toks#id(w_52)">
una</mw>
<mw id="mw_51" pos="I" mfeats="NN"
lemma="ehm" sfeats="intj"
href="toks#id(w_53)"> ehm</mw>
<mw id="mw_52" pos="PU" sfeats="par"
href="toks#id(w_54)"> -</mw>
<mw id="mw_53" pos="PU" sfeats="overlap"
href="toks#id(w_55)"> #</mw>
<mw id="mw_54" pos="D" mfeats="fs" lemma="un"
sfeats="art" sems="ind" href="toks#id(w_56)">
una</mw>
<mw id="mw_55" pos="N" mfeats="fs"
lemma="bandierina" sfeats="n" sems="com"
href="toks#id(w_57)"> bandierina</mw>
<mw id="mw_56" pos="PU" sfeats="punto"
href="toks#id(w_58)"> .</mw>
```

2. Incremental Shallow-to-Deep Parsing

Shallow or partial parsing produces minimal and incomplete syntactic structures, often in an incremental descriptive schema. In order to repeat some if not all of the features successfully analysed by full GETARUNS, we need to extend shallow parsing to deeper language analysis, while preserving robustness. In order to tackle deeper linguistic aspects we assume the following are essential requisites to fulfil:

- structural information must be extended in order to recover clause-level structure safely;
- lexical information should be tapped in order to help differentiate arguments from adjuncts; i.e. the lexicon should contain full subcategorization frames for most if not all verb, adjective, noun predicates that require them;
- grammatical functions should also be mapped onto the syntactic representation in order to take advantage of fundamental distinctions these descriptions afford: predicative vs. non-predicative functions are distinguished thus allowed a correct semantic mapping to take place.

As in most robust parsers, we use a sequence or cascade of transducers: however, in our approach, since we intend to recover sentence level structure, the process goes from partial parses to full sentence parses. Sentence and then clause level is crucially responsible for the right assignment of arguments and adjuncts to a

governing predicate head. This is clearly paramount in our scheme which aims at recovering predicate-argument structures, besides performing a compositional semantic translation of each semantically headed constituent.

So the first parser receives the input sentence split by previous processors, which is recursively/iteratively turned into a set of non-sentential level syntactic constituents - some of which can incorporate a PP headed by "of". Other operations solved at constituent level is that of collecting under the same constituent structure head level coordinate structures separated by "and/or".

Non-sentential level constituents, can be interspersed by heads which are beginning subordinate clause markers, like subordinating conjuctions, or parentheticals - by punctuation, indirect interrogative clauses - by interrogative pronouns. The final output is a list of headed syntactic constituents which comprise the usual set of semantically translatable constituents, i.e., ADJP, ADVP, NP, PP, VC (Verb Cluster). In addition to that, sentence level markers interspersed in the output are the following:

- FINT, interrogative clause marker;
- DIRSP, direct speech clause marker;
- FP, parenthetical clause marker;
- FC, coordinate clause marker;
- FS, subordinate clause marker;
- F2, relative clause marker.

The task of the following transducer is that of collapsing into the corresponding clause the clause material following the marker up to some delimiting indicator that can be safely taken as not belonging to the current clause level. In particular we assume that at each sentence level only one VCluster can appear: we define the VC as IBAR indicating that there must be a finite or tensed verb included in it. VClusters containing non-tensed verbal elements are all defined separately,

- SV2, for infinitive VCs;
- SV5, for gerundive VCs;
- SV3, for participial VCs.

The second transducer has also two additional tasks: it must take care of ambiguity related to punctuation markers such as COMMA, or DASH, which can either be taken as beginners of a parenthetical or indicators of a list, or simply as separators between main clause and subordinate/coordinate clause. It has also the task of deciding whether conjunctions indicated by FC or by FS are actually starting a clause structure or rather an elliptical structure.

The third pass is intended to produce an improvement on the sentence-level full parse, by transducing each constituent label into a corresponding grammatical function label. The rules are the following, and are taken from the inventory LFG theory and follow its rules and principles. In order to account for the ambiguous labelling of NPs, we use a logical flag associated to IBAR: it is set to false at the beginning of the parser; when the first NP is met and ibar(false) has success, it will be turned into SUBJ. When the IBAR is taken the flag is set to true so that the following NP will be turned into OBJ. We also compute another important feature of IBARs: their passivity. So whenever a passive IBAR is taken, we do not expect a following NP to belong to that clause level, but rather to the following one. Grammatical functional labels are then the following:

- ADJPs are turned into ACOMP;
- ADVPs are turned into ADJ;
- NPs are turned into SUBJ, in case the ibar flag is set to false; and into OBJ in case the ibar flag is set to true;
- PPs are turned into OBL;
- SV2, SV5, SV7, are all turned into VCOMP.

Some of these functional labels may undergo further changes when subcategorization is looked up in the lexicon: in particular,

- OBJs may become NCOMP;
- OBLs may become PCOMP;
- ADJs may become ADVCOMP.

Finally the fourth pass has the task of splitting complex sentences into simplex ones, or clauses. This may require recovering IBAR and complement structures following a relative clause or a subordinate clause functioning as noun complement, and rejoining it to its subject while preserving control information. As the previous ones, this level may lead to failures, which is recovered by simply considering all functions as belonging to the same clause and using IBARs as filters, by means of subcategorization.

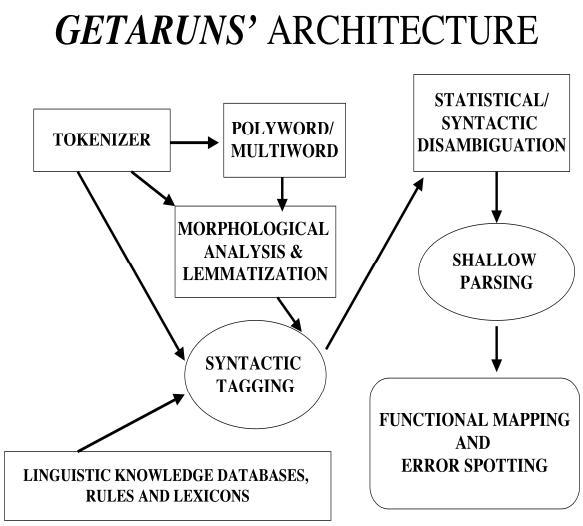

Fig. 1 GETARUNS Robust Parser Architecture

The output of the four transducers is passed to the algorithm that takes care of the creation of predicate-argument structures which has the additional task of taking into due account interclausal relations. To do that, semantic indices of governing predicates are used to assert dependencies between two adjacent clauses. This may also apply to a main clause and a clause-like adjunct like a gerundive or a participial.

3. Overlaps

Overlaps may be defined as a speech event in which two people speak at the same time by uttering actual words or in some cases non-words, when one of the speakers, usually the one which is not the current turntaker, interrupts or backchannels the current speaker. This phenomenon takes place at a certain point in time where it has to be anchored to the speech signal; but in order to be fully parsed and subsequently semantically interpreted, it needs to be referred semantically both to a following turn and to the local turn where it may produce conversational moves to repair what has been previously said by the current speaker.

One of the distinctive characteristics of naturalistic conversation (in contrast to monolog situations) is the presence of overlapping speech. Overlapping speech may be of several types, and affect the flow of discourse in various ways. An overlap may help to usurp the floor from another speaker (e.g., interruptions), or to encourage a speaker to continue (e.g., back channels), or simply end up just in an attempt at usurping the floor without success (Vain Interruption as defined by Bazzanella). In our work we have explored types of overlaps and their physical parameters, including prosodic aspects. As a preliminary and tentative definition we may define an overlap as being normally a physical event that happens in a single time unit in which two or more speakers want to comunicate different and non-coincident comunicative intentions. Exception made for rare cases in which the two or more speakers intended to say the same thing in the same time unit.

Speaker overlaps, are directly observable in our data, since by definition overlaps occur at points of simultaneous speech on more than one of the (individually recorded) channels, besides their explicit indication in the ortho-phonetic transcription thus transliterated into the orthographic transcription. What we are interested in is finding out whether there is any correlation between the onset of overlaps and their possible characterization from the point of view of syntactic structure, which we have proposed to treat by introducing a node of discourse constituency called OVL (overlap), from where the two temporally aligned components of overlapping, the overlappee and the overlapper stretch of speech/text, branch. The typologies proposed in the English literature and those suggested by Bazzanella will be verified in relation to their treatment at the level of syntactic constituency. Both punctuation and overlap have been discussed in the literature as correlating with prosodic cues. For example, past computational work has discussed prosodic features for sentence boundaries as well as disfluency boundaries. Past work in conversation analysis, discourse analysis, and linguistics has shown prosody to be a useful cue in turn-taking behavior.

3.1 Overlaps: why caring about them in the first place?

Why detecting and labeling Overlaps is so important? These are the most important reasons for taking care of them:

- They are very frequent;

- They may introduce linguistic elements which influence the local context;
- They may determine the interpretation of the current utterance;

and for these reasons,
they cannot be moved to a separate turn because they must be semantically interpreted where they temporally belong.
After moving overlaps to their original temporal position, as a side-effect, some turns are just empty conversational moves because the speaker has already been taking the turn with a previous overlap which may have been followed by a repairing move of the other speaker thus conversationally concluding the communicative exchange.

Tab 1. Overlaps data in Avip/Api Dialogues

1110 overlaps distributed over 20 files for a total of 4747 turns.
Turns with more than one overlap at their internal = 60
On average one overlap every 4.2 turns

As shown above, overlaps are very frequent indeed. Also consider the fact that total number of tokens for AVIP/API is 56,337, of which 18,710 are constituted by punctuation and turn tokens, and the remaining 37,627 tokens are words, quasi-words and interjections. If we divide up these total word related tokens by the number of turns we end up with an average of about 8 words per turn. Thus, one overlap every 35 words.
We also parsed another spontaneous spoken corpus corpus, the IPAR corpus, but only partially. For the sake of evidence, we also report preliminary data from this one which uses the Differences protocol: the main distinguishing feature of this corpus is the fact that the two speakers have no predefined role in the conversation so that turn-taking is much more independent and spontaneous, and is eventually only based on each speaker's attitude and personality. As can be gathered from the data on overlaps, in this case the number is almost doubled.

Tab 2. Overlaps data in IPAR Dialogues

424 overlaps distributed over 979 turns
Turns containing more than one overlap are 38.
On average one overlap every 2 turns

3.2 Overlaps and orthography: realignment and time irreversibility

In the original MapTask overlaps over two consecutive turns were simply marked off in blue colour: the words in blue overlapped. However, whenever there was more than one overlap in a single turn things became unclear, as shown in the following two examples taken from the materials made available on the web:

Dialogue 1.
FOLLOWER: what finish ?
GIVER: at the ch- at the chestnut tree.
FOLLOWER: right.

where we can surmise that "what finish" uttered by the Follower overlapped with "at the ch-" uttered by the Giver; then we are also led to believe that "tree" uttered by the Giver overlaps "right" uttered by the Follower and reported in the following turn. In this case no special problem seems to arise in linking the portion of overlapping materials. But consider now the following fragment:

Dialogue 2.
GIVER: no do-- all right okay, we'll we'll forg--.
FOLLOWER: I'm going I'm going right... I'm going right towards the yacht club?
GIVER: we'll forget about the yacht club just now.

Here we are led to consider the Giver's "no do" to overlap with something previously pronounced by the Follower: on the contrary, this overlaps with the Follower's "I'm going"; then the Giver's "okay, we'll we'll forg—." Overlaps with the Follower's "I'm going right". Finally the Follower's "yacht club" overlaps with the Giver's "we'll forget". As can be gathered, there is no real motivation of separating turns which have strictly interconnected materials apart from the need to have a linear description. And as a matter of fact linearizing in the case of overlaps is twice wrong: phenomena which should belong to one and the same time unit are represented by the orthography as belonging to two separate

time units. The colour is then used to rescue the temporal dimension.
A syncronic view of the phenomenon should result in a more compact and nonlinear way to use the orthography: for instance, Dialogue 1 could be written like this:

Dialogue 1.1
FOLLOWER: what finish ?/GIVER: at the ch-
GIVER: at the chestnut tree/FOLLOWER: right.

In this way, instead of artificially splitting the conversational moves into three turns, as a result of compacting overlapping portions of speech, only two turns would be represented by the orthography. We apply the same procedure to Dialogue 2:

Dialogue 2.1
GIVER: no do--/ FOLLOWER: I'm going
FOLLOWER: I'm going right...
GIVER: all right okay, we'll we'll forg--
./FOLLOWER: I'm going right
FOLLOWER: towards the yacht club?/GIVER: we'll forget
GIVER: about the yacht club just now.

but in this case the number of turns is almost doubled in order to capture overlapped portions of speech adequately.
The decisions taken in the Italian MapTask was to follow the original transcription schema and conventions: in particular, overlaps are fully marked in the local speech aligned orthographic transcription, by introducing the index of the turn containing the overlapping material, which however is not visible and should be looked up in the following turn. In addition, two #s are introduced at the front of the turn index and at the end of the overlapped speech as shown in the following example:

Dialogue 2.
p1#94: no <sp> cioè sì c'ha<aa> <mh> <sp> una specie di tappo
p2#95: sì #<p1#96> c'ha un ta+ tappo <sp># , sì
p1#96: #<p2#95> di funghetto# <lp> c'ha prima una base un po' altina

Dialogue 2.1
p1_94: no, cioè sì c'ha, una specie di tappo.
p2_95: sì ov_42 di funghetto < c'ha un ta_ tappo - >, sì.
Turn 95 contains an overlap which is introduced and erased from the following turn and indexed as shown in 4.1 version of the dialogue: the convention being that the ov_42 index is followed by the overlapper's speech intruding in the overlappee's turn. The material being overlapped then follows the open '<' and the close of the overlap is marked by the closing '>'. In this way the orthography linearizes the bidimensional event of the overlap by keeping the linguistic material within the same turn as adjacent text rather than scattering it in different turns. The ownership of the material by one of the speakers is guaranteed by its local respective position within the boundaries of the overlap: the ov_N starting symbol and the '>' at the end. It is important to notice that the two words are respectively pronounced by a woman and a man, the intruder utters with a rising tone: the implicit communicative intention is that of producing a better indication of the shape of the object currently under discussion and trying to get the other speaker to accept it.
The utterance contains a short pause <sp> right after the overlap which is then followed by an affirmative interjection "sì"/yes: this is a very common feature of overlaps in our corpus, a confirmation is a conversational act reacting to the overlapping material, which however is not present in the current utterance since it has been moved to the following turn. As can be understood by recomposing the overlapping portions of this conversation, what really happens is that the two speakers, Speaker ! and Speaker 2 are interacting very closely while the description of the scenario is carried on. At the same time at which a certain shape is individuated and properly described a consensus is reached: but this is reached by trial and errors in a continual re-approximation of the task. So a better linearization of the overlapped portions of conversation would result in the following orthography:
There are two internal repairs caused by the overlap: the first one is "sì"/Yes as a reaction of Speaker 2 to a first definition of the shape "tappo"/cork, which is however taken only as being suggestive "una specie di"/a kind of, of a better yet to be defined final shape. And in the

Speaker 2 turn, the repetition of “tappo” which is intentionally interrupted by recovering the turn role and suggesting the most appropriate shape, “di funghetto”/of a little mushroom.

So the new reconstructed splitting of the two turns better represents conversational moves and dialogue structure recovers linearity.
Here below we report the overlapped portion of dialogue we have been discussing in xml format as generated by our mapping algorithms.

```
<overlap  ov_42>
<w id=" ortsovr_1333 ortofon_tok=" # " href=" dgtdb04r_ortofonfile #id( ortfon_1277 )">  ov_42 </w>
<w id=" ortsovr_1334 ortofon_tok=" <p1#96> " href=" dgtdb04r_ortofonfile #id( ortfon_1278 )">  nil </w>
<w id=" ortsovr_1335 ortofon_tok=" c' " href=" dgtdb04r_ortofonfile #id( ortfon_1279 )">  nil </w>
<w id=" ortsovr_1336 ortofon_tok=" ha " href=" dgtdb04r_ortofonfile #id( ortfon_1280 )">  di </w>
<w id=" ortsovr_1337 ortofon_tok=" un " href=" dgtdb04r_ortofonfile #id( ortfon_1281 )">  funghetto </w>
<w id=" ortsovr_1338 ortofon_tok=" ta+ " href=" dgtdb04r_ortofonfile #id( ortfon_1282 )">  > </w>
<w id=" ortsovr_1339 ortofon_tok=" tappo " href=" dgtdb04r_ortofonfile #id( ortfon_1283 )">  nil </w>
<w id=" ortsovr_1340 ortofon_tok=" <sp> " href=" dgtdb04r_ortofonfile #id( ortfon_1284 )">  nil </w>
<w id=" ortsovr_1341 ortofon_tok=" # " href=" dgtdb04r_ortofonfile #id( ortfon_1285 )">  c' </w>
<w id=" ortsovr_1342 ortofon_tok=" , " href=" dgtdb04r_ortofonfile #id( ortfon_1286 )">  ha </w>
<w id=" ortsovr_1343 ortofon_tok=" sì " href=" dgtdb04r_ortofonfile #id( ortfon_1287 )">  un </w>
<w id=" ortsovr_1344 ortofon_tok=" p1#96: " href=" dgtdb04r_ortofonfile #id( ortfon_1288 )">  ta_ </w>
<w id=" ortsovr_1345 ortofon_tok=" # " href=" dgtdb04r_ortofonfile #id( ortfon_1289 )">  tappo </w>
<w id=" ortsovr_1346 ortofon_tok=" <p2#95> " href=" dgtdb04r_ortofonfile #id( ortfon_1290 )">  , </w>
<w id=" ortsovr_1347 ortofon_tok=" di " href=" dgtdb04r_ortofonfile #id( ortfon_1291 )">  # </w>
<w id=" ortsovr_1348 ortofon_tok=" funghetto " href=" dgtdb04r_ortofonfile #id( ortfon_1292 )">  - </w>
<w id=" ortsovr_1349 ortofon_tok=" # " href=" dgtdb04r_ortofonfile #id( ortfon_1293 )">  sì </w>
</overlap>
```

Tab3. Overlapped portion of Dialogue with hrefs to ortho-phonetic transcription

3.4 Overlaps and syntax

As said above, overlaps challenge all criteria of linguistic representation which require the input sentence to be mono-dimensional, i.e. to contain the utterance of one single speaker. This fact is semantically essential in order to guarantee the linguistic representation to be interpretable. On the contrary, overlapped linguistic material, i.e. sentences which contain at the same time linguistic material coming from two or more participants in the dialogue are not only hard to parse: they might also constitute an obstacle to semantic interpretation. Consider the previous example Dialogue 2, Speaker 2 utterance, whose syntactic structure is reported below,

Dialogue 2.b
da(turn(p2_95),cp(intj(si'), ovl(overlap(ov_42), spd(pd(di), sn(n(funghetto))), par(<), f(ibar(expl(c), vc(ha)), compc(sn(art(un), abbr(ta_), sn(n(tappo))))), par(par), overlap(>)), punt(virg), cp(intj(si'))), punto(.))

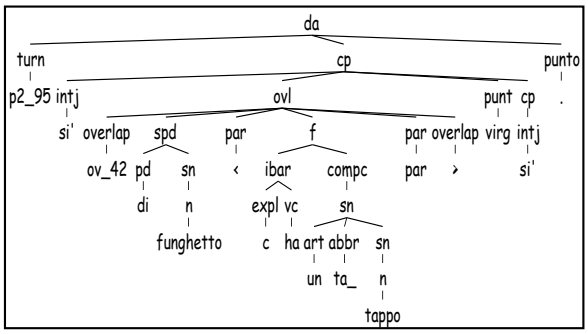

Fig. 2 Syntactic Structure for Dialogue 2.b with temporally aligned overlap

The realignment of all turns has given as a result a certain number of empty turns, i.e. all those turns which had been artificially built by simply containing overlapping material which had been already uttered by the current speaker before the previous turn was elapsed.
The need to represent linguistic information related to two speakers in the same syntactic structural representation, which is both semantically and pragmatically strongly intertwined has a lot of theoretical implications.

This implements principles of linguistic representation expressed in previous work of ours, in particular in Delmonte, (1987), where syntactic structure was to interact with semantic and pragmatic structure in order to take into due account phenomena like Contrastive and Emphatic Focus. It is generally agreed that, in studying the facts of language, two domains have been devised in the process of defining grammatical relations.

a. *Sentence Grammar* which encompasses phenomena belonging to clause level, with NP and S (or S') as the relevant domains in which to specify syntactic constraints and dependancies (Rizzi, 1982). This theoretical abstraction is so restricted to be able to account for basic facts of competence in language acquisition;

b. *Discourse Grammar* which is crucially grafted onto rules of sentence grammar; it does not directly relate to unconscious and innate LAD mechanisms but stems and develops on extralinguistic, contextual/situational or pragmatic conditions.

As a matter of fact, no neat division should be drawn between these two theoretical domains, apart from empirical reasons, i.e. in order to reduce interfering factors which do not contribute in an essential way to the construction of an internal grammar. In particular, the realm of performance, being the less studied if compared to competence, contains quite a number of such interfering factors. We might also surmise that a lot of performance (as such describable within a discourse grammar) interferes strongly with competence (Bresnan, 1982: xxiii) leading to an interactive (see Marsley-Wilson, Tyler, 1980), model for discourse understanding, rather than a sequential one.

Interpretation could be triggered independently from sentential material or be determined by the presence of coreferring extrasentential expressions; as a further option, it could be triggered locally by logical operators which in turn may vary their scope according to the presence of extrasentential factors.

In other words, to allow for feedback to take place between the two levels of grammatical relations, we need discourse level phenomena to be adequately represented by sentence grammar. This is certainly the case with the case we are tackling now: overlaps take place at a discourse level, however their import is deeply grafted into sentence grammar, by conditioning interpretation from taking place. Coming now to our corpus, where as said above overlaps are on average occuring 1 every two turns, we have been able to detect their internal structure by means of syntactic annotation to be as follows:

- containing linguistic material which has started on their left;
- containing linguistic material which continues on their right;
- containing linguistic material which only weakly is related on their left or on their right;
- containing linguistic material which does not relate with the context;

The most numerous group is constituted by group 4 and group 1 with 540 occurrences; group 2 contains 100 overlaps; group three 310 overlaps. Group 1 and 2 are certainly the most interesting groups to study. From a strictly syntactic point of view, overlaps may interrup constituents but also internal sentences within complex sentences that contain them. Looking into these in more detail, we have found that:

- 330 are cases of constituent interruption;
- 210 are cases of interruption at higher than constituent level;

Interruption intervening between specifier and head, as well as between preposition and NP are treated separately.

Another interesting example is represented by the following utterance, where the overlapper corrects the current speaker – the overlappee – who, as a consequence of that, interrupts its utterance and confirms what the overlapper said.

“eeh, la spalla sinistra del bambino è leggermente più ov_73 alta della destra, sì > alta del, sì <.”/the left shoulder of the child is slightly more ov_73 high than the right one, yes > high of the, yes <.”

Whose syntactic structure is,

cp(intj(eeh),f(sn(art(la),n(spalla),sa(ag(sinistra)),spd(partd(del),sn(n(bambino)))),ibar(vc(e')),sq(savv(avv(leggermente)),in(piu')),ovl(overlap(ov_73),f3(sa(ag(alta),spd(partd(della),sn(ag(destra))))),punt(virg),cp(intj(si')),par(>),sa(ag(alta),spd(partd(del))),punt(virg),cp(intj(si')),overlap(<))), punto(.))

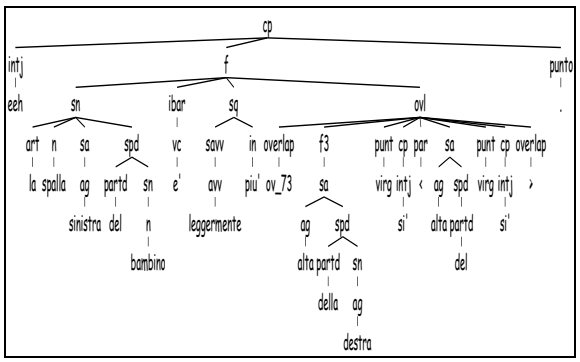

Fig. 3 Syntactic Structure for Dialogue 3 with temporally aligned overlap and linguistic material on the right

In the following, we will be concerned only with a subsection of our corpus – IPAR Dialogues, called Differences Dialogues, which contains some 5000 tokens. From syntactic analysis we have built 385 autonomous utterances, 336 thereof coinciding with actual turns and 49 derived from the insertion of punctuation in the conversation. The analysis has also produced 147 overlaps which will be commented in detail below.

The treebank has the following constituents:

- 588 NP (nominal phrases)
- 91 AP (adjectival phrases)
- 132 PP (prepositional phrases)
- 72 PPOF (prepositional phrases with preposition OF and its amalgamated forms)
- 27 PPBY (phrases preposizionali con preposizione BY and its amalgamated forms)
- 77 AVVP (adverbial phrases)
- 34 QP (quantified phrases)
- 3 VPINF (infinitivals)
- 4 VPPART (participials)
- 6 VPGER (gerundives)
- 27 S2 (relative clauses)
- 13 SC (sentential complements)
- 257 SF (sentential fragments)
- 134 COORDS (coordinate sentences)
- 7 SUBS (subordinate sentences)
- 267 S (simplex sentences)
- 81 INTS (esclamatives and simplex interrogatives)
- 18 SPAR (parentheticals)
- 273 IBAR (verbal structures)
- 41 IR_INFL (verbal structures unreal mood/tense)
- 369 CP (complex sentences)
- 28 CP_INT (esclamatives and complex interrogatives)
- 134 COMPC (copulative complements)
- 34 COMPIN (intransitive complements)
- 51 COMPT (transitive complements)
- 19 COORD (non sentential coordinate structures)

Overall, the treebank contains 1586 non sentential constituent labels, 1200 sentential constituent labels. The total sum of syntactic constituent labels amounts to 2787. To these canonical constituent labels we should than add non canonical ones, marking dialogue level phenomena, i.e. DA (dialogue actions) and OVL (overlap) which amount respectively to 336 and 147. In sum we come up with 3269 constituent labels.

Some interesting remarks may be the number of copulative predications, 134, very high if compared to the other types of complement structures. Then the number of sentence fragments 104, almost the same number of interrogatives/exclamatives.

Coming now to overlaps, which as said above are on average occuring 1 every two turns, have been organized as follows:

- containing linguistic material which has started on their left;
- containing linguistic material which continues on their right;
- containing linguistic material which only weakly is related on their left or on their right;
- containing linguistic material which does not relate with the context;

The most numerous group is constituted by group 4 and group 1 with 54 occurrences; group 2 contains 10 overlaps; group three 31 overlaps. Group 1 and 2 are certainly the most interesting groups to study. Overlaps may interrup constituents but also internal sentences within complex sentences that contain them. Looking into these in more detail, we have found that:

- 33 are cases of constituent interruption;
- 21 are cases of interruption at higher than constituent level;

Interruption intervening between specifier and head, as well as between preposition and NP are treated separately.

Abbiamo considerato interruzione di costituente casi di separazione tra il contenuto dello specificatore e la testa, oppure come nei SP tra la preposizione e il SN che riportiamo qui in basso:

da-[turn-p2_39, cp-[par-(-), f3-[intj-eeh], f3-[abbr-il_], punt-',', sp-[p-in, sn-[deit-quella, f2-[rel-che, f-[ibar-[vc-sta], compc-[intj-ehm, sp-[p-per, sn-[n-terra]]], punt-',', spd-[pd-di, ovl-[overlap-ov_19, cp-[intj-sì], par->, sn-[n-palla]], overlap-#]]]], punto-.]

f-[ibar-[clitabl-ci, vc-sono], compc-[sn-[num-due, n-linee, fp-[punt-',', abbr-se_, punt-','], f2-[rel-che, if-[bar-[vin-partono], spda-[pda-da, sn-[dim-questa, spda-[pda-da, sn-[dim-questa, ovl-[overlap-ov_20, cp-[f2-[rel-che, ibar-[vin-escono]], punt-',', savv-[avv-fuori]], par->, sn-[n-palla]]]]]]], overlap-#]]]], punto-.]

cp-[f-[ibar-[expl-c, vc-ha], punt-',', f3-[abbr-fini_]], punt-',', fc-[congf-cioè, f3-[sn-[art-la, abbr-part_, abbr-fini_]]], punt-',', cp-[intj-sì, f3-[sn-[art-la, abbr-pa_, sn-[art-la, n-parte, ag-finale]]]], cp-[intj-sì, f-[ibar-[vc-è, compc-[sp-[p-a, ovl-[overlap-ov_51, f3-[intj-eh], par->, sn-[n-punta]], overlap-#]]]]], punt-',', f-[ibar-[expl-c, vc-ha], compc-[fc-[ccom-come, conjl-come_se, f-[ir_infl-[vcir-fossero], compc-[sn-[num-due, n-dita, sp-[part-alla, sn-[n-fine]]]]]]]], punto-.]

cp-[f3-[dim-quel], punt-',', f3-[sa-[ppas-attaccato, sp-[part-al, sn-[n-bordo, cp-[intj-no], ovl-[overlap-ov_58, cp-[intj-sì, sv3-[ppas-attaccato, comp-[sp-[part-al, sn-[n-bordo]]]]], par->, spd-[partd-della, sn-[n-figura]]]]], overlap-#], punto-.]

cp-[intj-eeh, f-[sn-[art-la, n-spalla, ag-sinistra, spd-[partd-del, sn-[n-bambino]]], ibar-[vc-è], sq-[savv-[avv-leggermente], in-più], ovl-[overlap-ov_73, f3-[sa-[ag-alta, spd-[partd-della, sn-[ag-destra]]]], punt-',', cp-[intj-sì], par->, sa-[ag-alta, spd-[partd-del]], punt-',', cp-[intj-sì], overlap-#]], punto-.]

da-[turn-p1_118, fc-[congf-cioè, cp-[f3-[sn-[num-due, n-dita], punt-',', sn-[num-due]], intj-ehm, punt-',', intj-no, punt-',', fc-[congf-ma, f-[ibar-[neg-non, vt-penso], compt-[fac-[ir_infl-[clitabl-ci, vcir-sia], compc-[savv-[avv-là], sn-[art-'a/', ovl-[overlap-ov_52, cp-[intj-no], par->, sn-[n-cosa], overlap-#]]]]]]]], punto-.]

da-[turn-p1_214, fc-[congf-poi, f-[ibar-[clitabl-ce, vc-sta], compc-[sn-[art-una, ovl-[overlap-ov_95, cp-[intj-sì], par->, sn-[n-lineetta]], overlap-#]]]], punto-.]

da-[turn-p1_22, cp-[intj-no, punt-',', compt-[sp-[p-per, sn-[pron-me]], f-[ibar-[vc-è], compc-[sp-[p-verso, ovl-[overlap-ov_11, congf-e, congf-allora, par->, sn-[n-sinistra], overlap-#]]]]]], punto-.]
da-[turn-p1_250, fc-[cong-sia, f3-[sp-[p-a, sn-[n-destra]]], cong-che, sp-[p-a, ovl-[overlap-ov_115, fc-[cong-che, f3-[abbr-s_]], par->, sn-[n-sinistra]], overlap-#]], punto-.]

da-[turn-p1_252, cp-[par-(-), f-[ibar-[vc-è], compc-[sa-[in-più, ovl-[overlap-ov_117, cp-[intj-sì], par->, sa-[ag-corto], overlap-#], par-(-), ovl-[overlap-ov_118, f-[ibar-[vt-facciamo, compt-[savv-[avv-così]]]], par->, f3-[intj-mh], overlap-#]]]]], punto-.]

da-[turn-p1_265, fint-[int-quale, ovl-[overlap-ov_120, f3-[sn-[art-la, n-nuvola]], par->, sn-[n-nuvola]], overlap-#], puntint-?]

da-[turn-p1_284, cp_int-[f3-[sn-[pron-altra], punt-',', sn-[pron-altra], punt-','], fint-[f3-[sa-[deit-quella, ovl-[overlap-ov_129, fs-[cosu-se, f-[sn-[pron-tu], ibar-[vin-parti]]], par->, sa-[in-più, ag-grande]]], overlap-#]]], puntint-?]

da-[turn-p1_288, f-[ir_infl-[virin-guarda]], punt-',', par-(-), f-[ibar-[vt-tocca], compt-[sq-[art-un, q-pô], sn-[art-i, n-capelli, spd-[partd-del, ovl-[overlap-ov_131, cp-[intj-no], par->, sn-[n-bambino]]]], overlap-#]], punto-.]

da-[turn-p1_332, cp-[intj-no, fp-[punt-',', congf-allora, punt-','], f-[ir_infl-[virt-aspetta], fp-[punt-',', congf-allora, punt-',']], f-[coord-[sn-[num-uno, num-due, num-tre], par-(-), sn-[num-cinque, num-sei, num-sette], ovl-[overlap-ov_146, sn-[num-otto], punt-','], cp-[intj-benissimo], par->, sn-[num-otto], overlap-#], fc-[congf-e, congf-però, f-[ibar-[neg-non, vt-conto], compt-[sn-[deit-quella, sa-[ag-dritta]], punt-',', sp-[part-all, sn-[n-orizzonte]]]]]], punto-.]

da-[turn-p1_68, cp-[intj-sì], punt-',', f3-[sq-[q-uno], ovl-[overlap-ov_32, f3-[sq-[q-uno, in-solo]], par->, sq-[in-solo]], overlap-#], punto-.]

da-[turn-p1_90, f-[ir_infl-[virt-vediamo], compt-[sq-[art-un, ovl-[overlap-ov_41, f3-[art-la], par->, q-pô], overlap-#]]], punto-.]

da-[turn-p2_161, f3-[coord-[sa-[ag-grande], punt-',', sn-[num-una, sa-[ag-piccola]], punt-',', sn-[num-una, ovl-[overlap-ov_70, cp-[intj-sì], par->, sa-[ag-grande]], overlap-#]]], punto-.]

da-[turn-p2_183, cp_int-[f-[ibar-[clit-si, vt-vede], compt-[savv-[in-solo], sn-[art-il, ovl-[overlap-ov_79, cp-[intj-sì], par->, n-pollice], overlap-#]]]], puntint- ?]

da-[turn-p2_225, f3-[spd-[partd-della, ovl-[overlap-ov_101, cp-[intj-sì], par->, sn-[n-palla]]], overlap-#], punto-.]

da-[turn-p2_271, f3-[spd-[pd-di, sn-[n-metà, spd-[partd-della, ovl-[overlap-ov_122, cp-[intj-sì], par->, sn-[n-bandierina]]], overlap-#]]], punto-.]

da-[turn-p2_275, f3-[spd-[partd-della, ovl-[overlap-ov_125, cp-[intj-sì, intj-sì, intj-mh, f3-[savv-[avvl-più_o_meno], sn-[n-metà]]], par->, spd-[partd-della, sn-[n-nuvola]]]], overlap-#], punto-.]

da-[turn-p2_299, f-[ibar-[vc-sta], compc-[sp-[part-all, sn-[n-altezza, spd-[partd-dei, ovl-[overlap-ov_136, cp-[sv3-[ppas-capito], congf-cioè, sp-[part-nel, sn-[n-senso]]], par->, sn-[n-capelli, spd-[partd-del, sn-[n-bambino]]]], cp-[intj-sì], overlap-#]]]]], punto-.]

da-[turn-p2_307, f-[ir_infl-[virt-segui], par-(-), compt-[sn-[art-il, poss-mio, ovl-[overlap-ov_139, cp-[intj-sì], par->, sn-[n-ragionamento]], punt-',', congf-allora], overlap-#]], cp-[spda-[partda-dalla, spda-[partda-dal, sn-[n-bordo, sa-[ag-superiore, spd-[partd-della, sn-[n-figura]]]]]]], punto-.]

da-[turn-p2_313, f-[ibar-[vin-inizia], sn-[art-la, n-punta], compt-[sp-[p-a, sn-[art-un, sa-[ag-certo], ovl-[overlap-ov_140, cp-[intj-sì], par->, sn-[n-punto]], overlap-#], fp-[punt-',', f3-[intj-eh], punt-','], sp-[p-tra, sn-[art-il, n-bordo, ag-superiore, spd-[partd-della, sn-[n-figura, f2-[rel-dove, ibar-[clit-si, abbr-inc_], fp-[punt-',', f3-[partd-della], punt-','], f2-[rel-dove, ibar-[clit-si, vin-incrocia], sn-[art-la, n-nuvola]]]]]]]]], punto-.]

da-[turn-p2_327, f-[sn-[num-uno], ovl-[overlap-ov_144, fc-[congf-quindi, f-[ibar-[vc-sono], compc-[sn-[coord-[num-uno, num-due, num-tre], punt-',', sn-[num-sette, num-otto]], par->, sn-[num-due, num-tre, num-quattro, num-cinque, num-sei, num-sette, num-otto], overlap-#]]]]], punto-.]

da-[turn-p2_331, f-[ibar-[neg-non, vit-contare], compt-[sn-[art-la, n-linea, ag-continua], f3-[partda-dalla], f3-[partda-dalla], cp-[intj-ehm, fc-[congf-insomma, spd-[partd-del, ovl-[overlap-ov_145, spd-[partd-della, sn-[n-spiaggia]], punt-',', f3-[intj-ah], par->, spd-[partd-della, sn-[n-battigia]], punt-',', cp-[intj-sì], punt-',', congf-insomma, overlap-#]]]]]], punto-.]

da-[turn-p2_35, cp_int-[f-[ibar-[neg-non, vt-conti], compt-[sn-[art-la, ovl-[overlap-ov_15, fc-[congf-cioè, par->, sn-[num-terza]]], overlap-#]]]], puntint- ?]

da-[turn-p2_79, f-[ibar-[vc-è], compc-[sa-[ag-chiuso], congf-comunque, sn-[art-il, ovl-[overlap-ov_38, cp-[intj-sì], par->, n-becco], overlap-#]]], punto-.]

da-[turn-p2_87, cp-[intj-sì, punt-',', savv-[avv-sotto], f-[coord-[ibar-[clitabl-ci, vc-sono]], punt-',', f-[ibar-[vin-escono], compin-[sp-[php-fuori_dalla, ovl-[overlap-ov_40, cp-[intj-no], par->, sn-[n-cosa]], spda-[partda-dal], overlap-#]]]]], da_riempire- §]

f3-[par-(-), sn-[num-una, num-due, num-tre], ovl-[overlap-ov_142, f-[ir_infl-[virt-aspetta]], par->, sn-[num-quattro, cong-e, num-cinque], overlap-#], punto-.]

fc-[congf-poi, par-(-), f-[ibar-[neg-non, clitabl-c, vc-è], compc-[savv-[avv-più], sn-[art-la, n-lineetta]]], punt-',', f-[ibar-[clitabl-c, vc-è], compc-[sn-[art-la, ovl-[overlap-ov_107, cp-[intj-sì], par->, n-mano]], overlap-#]], punto-.]

fc-[f-[sn-[art-la, num-prima], ibar-[vc-è], compc-[sq-[art-un, q-pô, sa-[in-più, ag-lunga], spd-[p-de, sn-[qc-tutte]]]]], punt-',', f3-[sn-[art-la, num-seconda], sq-[art-un, ovl-[overlap-ov_46, cp-[intj-sì, intj-sì, intj-sì], par->, sq-[q-pô, spd-[pd-di, par-(-), sq-[in-meno]]], overlap-#], punt-',', sq-[in-più, abbr-pi_], punt-',', cp-[intj-vabbè]]], punto-.]

In tutti questi casi, le sovrapposizioni possono costituire elementi di anticipazione di quanto il parlante sovrapposto aveva programmato mentalmente di comunicare, e quindi possono influire sul corso della enunciazione. Se notiamo però c'è un caso di riformulazione dello stesso concetto – spiaggia/battigia, nel turno p2_331.C'è poi un caso in cui entrambi i parlanti pronunciano la stessa parola, per cui la sovrapposizione si riduce a una conferma della stesso concetto – i parlanti pensavano alla stessa cosa, ed è il turno p2_101. Nel turno p2_171, che riportiamo qui in basso sembra invece che la sovrapposizione provochi un cambiamento di piano dal momento che il sovrapponente ha già espresso lo stesso contenuto linguistico che quindi risulta essere inutile ed è sufficiente riprenderlo con un "sì".

cp-[intj-eeh, f-[sn-[art-la, n-spalla, ag-sinistra, spd-[partd-del, sn-[n-bambino]]], ibar-[vc-è], sq-[savv-[avv-leggermente], in-più], ovl-[overlap-ov_73, f3-[sa-[ag-alta, spd-[partd-della, sn-[ag-destra]]]], punt-',', cp-[intj-sì], par->, sa-[ag-alta, spd-[partd-del]], punt-',', cp-[intj-sì], overlap-#]], punto-.]

Per i casi restanti sembra che il sovrapponente si limiti ad esprimere conferma "sì" o rifiuto "no" di quanto l'attuale padrone di turno sta enunciando, riservandosi poi di esprimere eventuali revisioni a quanto detto.

Per quanto riguarda invece gli esempi del secondo tipo di sovrapposizioni, quelle che contengono materiale che continua alla destra, che sono in numero notevolmente inferiore, si tratta quasi sempre di interruzioni di livello superiore, come si può vedere dagli esempi riportati qui di seguito:

da-[turn-p2_173, cp_int-[ovl-[overlap-ov_74, cp-[intj-sì, punt-',', f-[ir_infl-[virt-

```
vediamo], compt-[sq-[art-un, q-pô], fint-
[int-che, ibar-[clit-se, vsup-pò]]]]], par->,
fc-[congf-poi, f3-[intj-ah, sn-[art-le, n-
dita, overlap-#], spd-[coord-[partd-del, sn-
[n-bambino]], punt-',', spd-[partd-dei, sn-
[n-piedi]]]]]]], puntint- ?]

da-[turn-p1_226, par-'-', ovl-[overlap-
ov_102, cp-[f-[ibar-[clitabl-ce, clit-n, vc-
è], sn-[abbr-u_], par->, f-[ibar-[clitabl-c,
vc-è], compc-[sn-[art-una, n-linea, overlap-
'#']], sa-[in-più, ag-lunga], sp-[p-con,
savv-[avv-sopra], sn-[num-due, n-
lineette]]]]]], punto-.]

da-[turn-p2_245, ovl-[overlap-ov_113, cp-
[intj-sì], par->, f-[ibar-[clitabl-c, vc-è],
compc-[sn-[art-una, overlap-#], n-specie,
spd-[pd-di, sn-[n-arco]]]]], punt-,, cp-
[intj-giusto], puntint- ?]

da-[turn-p1_50, ovl-[overlap-ov_25, fc-
[congf-poi], da_riempire- §, par->, f-[ibar-
[expl-c, vc-ha], compc-[sn-[art-un, overlap-
#], n-segnetto], congf-poi, sp-[part-al, sn-
[n-petto]]]], punto-.]

cp-[ovl-[overlap-ov_63, f3-[intj-mh], par->,
fc-[congf-e, f-[ibar-[clit-si, overlap-#],
vin-stacca], compt-[congf-quindi, spda-
[partda-dal, sn-[n-bordo]]], fc-[congf-e,
congf-poi, sn-[num-due, sa-[ag-normali]]]],
punto-.]

da-[turn-p2_273, par-(-), ovl-[overlap-
ov_123, , cp-[intj-sì], par->,  f3-[sn-[art-
la, n-parte, overlap-#], sa-[ag-finale]]],
f3-[sn-[deit-quella, f2-[rel-che, f-[ibar-
[vt-tocca], compt-[f3-[art-la], punt-',', sp-
[part-sul, sn-[n-bordo]]]]]], punt-',', cp-
[intj-sì, punt-',', intj-sì, punt-',', f-
[ibar-[ausa-ho, vppt-capito]]], punto-.]


da-[turn-p2_101, cp-[intj-eh, f-[sn-[art-la,
sn-[art-la, n-vela]], ibar-[expl-c, vc-ha],
compc-[f3-[art-le], cp-[intj-ehm], f3-[art-
le], cp-[intj-ehm]]], punto-'.']
cp-[ovl-[overlap-ov_45, f-[ibar-[clitabl-ci,
vc-sono, compc-[sn-[q-delle, sn-[n-
lineette]]]], punt-',', cp-[intj-boh]], par-
'>', f-[sn-[n-lineette], overlap-'#'], ibar-
[vc-sono], compc-[sn-[num-cinque]]], punto-
'.']
da-[turn-p1_102, cp-[]]
```

Abbiamo riportato tutto il materiale linguistico che precede e segue il turno p2_101, overlap-ov_45 in cui la parlante donna mostra incertezza nel definire l'oggetto da sottoporre all'attenzione del suo interlocutore, il quale prontamente interviene e lo dice nello stesso momento in cui lo pronuncia anche la sovrapposta, "le lineette".
Bisogna dire che molto spesso le sovrapposizioni servono a confermare quello che il possessore di turno sta per dire o aveva intenzione di dire, come nel caso del turno p1_50, e dell' overlap-ov_25, oppure del turno p1_226 e dell' overlap-ov_102. Nel turno p2_173, overlap-ov_74 si ha l'impressione che la donna dopo aver sentito il materiale sovrapposto si ricreda e individui una ulteriiore differenza nella dita "del bambino dei piedi" come dice lei, cioè le dita dei piedi del bambino.
Una conferma che nella sezione precedente avviene durante la sovrapposizione e quindi il sovrapponente è comunque in grado di prevedere la continuazione dell'enunciato del sovrapposto non ancora completato come avviene nel turno p2_79, overlap-ov_38.
E' interessante notare quello che avviene al turno p2_331, overlap-ov_145, in cui il possessore di turno sta esprimendo un enunciato in forma di esortazione negativa, e il materiale sovrapposto esprime lo stesso concetto in una singola unità di tempo, ma con elementi linguistici differenti: "la spiaggia" il sovrapponente, l'uomo e "la battigia" il sovrapposto, la donna.
Una cosa diversa avviene nel turno p2_275, overlap-ov_125, in cui il possessore del turno parla di una dimensione spaziale riferita ad un oggetto specifico "il margine superiore della nuvola" e il sovrapponente la corregge indicando una posizione che si riferisce al luogo stesso ma specificandola ulteriormente "più o meno a metà".
Un altro caso di riformulazione a seguito di una sovrapposizione avviene nel turno p2_265, overlap-ov_120, in cui il possessore del turno chiede "a che altezza" stia la "nuvola rispetto alla barca" senza specificare di quale nuvola si tratti – ce n'è più di una.

```
da-[turn-p2_265, fint-[sp-[p-a, int-che, sn-
[n-altezza]], f-[ibar-[vc-sta], compc-[sn-
[art-la, n-nuvola], sp-[php-rispetto_alla,
part-alla, f3-[intj-mh], sp-[part-alla, sn-
[n-barca]]]]]], puntint-'?']
da-[turn-p1_265, fint-[int-quale, ovl-
[overlap-ov_120, f3-[sn-[art-la, n-nuvola]],
par-'>', sn-[n-nuvola]], overlap-'#'],
puntint-'?']
da-[turn-p2_267, f3-[sn-[deit-quella, spd-
[pd-di, sn-[n-destra]]]], punto-'.']
```

Il parlante si accorge di aver fornito una informazione insufficiente e cerca subito di riparare nello stesso momento in cui il suo interlocutore interviene per chiedere maggiori indicazioni sulla localizzazione della nuvola. A quel punto il parlante, sovrapposto pronuncia due frammenti di enunciato: il primo, "la nuvola" che

abbiamo marcato di rosso e poi il secondo frammento in un enunciato che abbiamo indicato in giallo che risponde alla domanda del suo interlocutore.

3.4 Overlaps, prosody and semantics

If utterances containing overlaps may be represented syntactically, then it should also follow that they should be interpretable semantically. In fact, this may be true for all overlaps containing linguistic material which is related to its context, right or left. However, when the linguistic material does not relate locally to any portion of the utterance it is hard to define a strategy. In the former case, the interpretation follows from a treatment of overlapped material as belonging to the current clause – as a fragment - or to a previous utterance – as an ellipsed fragment.

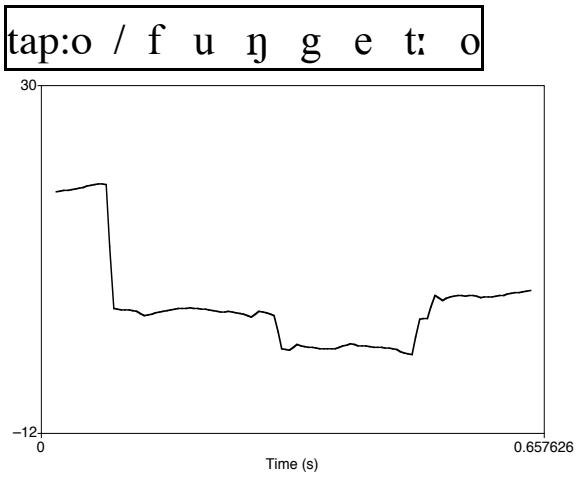

Fig.4 Overlapped intonation related to the two words“tappo-funghetto”/cork-small mushroom

To conclude, we analysed the prosodic content of those overlaps constituting interruption at constituent level and we found a strong correlation with the acoustic signal. In particular, at FØ level, it is usually the case that the two speakers are produced different intonational curves, such as the ones detected with the example examined above, which we report in Fig.4. The FØ refers to the two overlapping words “tappo/funghetto” cork/small mushroom, and the first word is present only with the last syllable “po”. It is important to notice that the two words are respectively pronounced by a woman and a man, the intruder utters with a rising tone: the implicit communicative intention is that of producing a better indication of the shape of the object currently under discussion and trying to get the other speaker to accept it. And the backchannel ending the communicative exchange testifies to that. In other words, the overlapper is introducing a fragment as a query which is contextually approved by the overlappee. From a pragmatic point of view the communication is now complete.
The semantics of this utterance can be represented in a flat logical form where the DA (Dialogue Act) contains an OVL to which the governing predicate “HAVE” is applied twice, as follows,

Dialogue 2.b
```
DA(turn(p2_95),
    OVL(prop(sì(y1),
        tappo(x1),
          AVERE(y1,x1)),
            prop(funghetto(x2),
              sì(y2),
                AVERE(y2,x2))))
```
OVL is thus treated as a modality operator which has scope over two propositions.

3.5 Overlaps and multilayered representation

All syntactic and lower levels representations are accessible from our website, http://project.cgm.unive.it clicking on “Progetto IPAR”. In particular the dialogues have been visualized by a Java program that maps on runtime the syntactic structures in a mirrored double window, where on the left side are the utterances of the text and on the right side the syntactic tree where overlaps are included, organized vertically. By clicking on the turn number one can connect to the orthographic original transcription we called “orthophonetic”, from where the acoustic speech files can be reached. This is done by moving into another window where the two transcriptions are comparable, the one with the overlaps assigned to a separate turn, and the other with the temporally

aligned overlaps. From this window the overlap itself can be reached and listen to.

4. References

Bazzanella, Carla 1994. "LE INTERRUZIONI", in *Le facce del parlare*, Cap 8, Firenze/Roma: La Nuova Italia, pp.176.

Bard, E.G., Anderson, A.H., Sotillo, C., Aylett, M., Doherty-Sneddon, G. & Newlands, A. (2000) Controlling the intelligibility of referring expressions in dialogue, *Journal of Memory and Language* , 42(1) , 1-22.

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

Bernsen, N. O., Dybkjaer, L. & M. Kolodnytsky 2002. The NITE Workbench - A Tool for Annotation of Natural Interactivity and Multimodal Data. *Proceedings of LREC 2002,* Las Palmas.

Delmonte R. 2003. Parsing Spontaneous Speech, in Proc. EUROSPEECH2003, Pallotta Vincenzo, Popescu-Belis Andrei, Rajman Martin "Robust Methods in Processing of Natural Language Dialogues" , Genève, pp, 16-23.

Delmonte R. 2000. Shallow Parsing And Functional Structure In Italian Corpora, LREC, Atene, pp.113-119.

Delmonte R. 2001. How to Annotate Linguistic Information in FILES and SCAT, in Atti del Workshop "La Treebank Sintattico-Semantica dell'Italiano di SI-TAL, Bari, pp.75-84.

Delmonte R. 1987. Focus and the Semantic Component, in *Rivista di Grammatica Generativa*, 81-121.

Flammia, G., & Zue, V. (1995). N.b.: A Graphical User Interface for Annotating Spoken Dialogue. In J. Moore, M. Walker, M. Hearst, L. Hirschman, & A. Joshi (Eds.), *Working Notes from the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation* (pp. 40-46). Palo Alto: AAAI.

Hindle D. 1993. "Deterministic parsing of syntactic nonfluencies", In *Proc. ACL*, pages 123-128.

Laprun, C., Fiscus, J. G., Garofolo, J., & Pajot, S. (2002, May). A Practical Introduction to ATLAS. *Paper presented at the 3rd International Conference on Language Resources and Evaluation (LREC),* Las Palmas.

Lickley,R., "Detecting Disfluency in Spontaneous Speech", *Thesis*, Department of Linguistics, University of Edinburgh 1994.

Ma, X., Lee, H., Bird, S., & Maeda, K. 2002. Models and Tools for Collaborative Annotation. *LREC 2002,* Las Palmas.

Marsley-Wilson W., Tyler L.K. 1980. The Temporal Structure of Spoken Language Understanding, Cognition 8, Elsevier Sequoia S.A.,Lausanne, 1-71.

McKelvie,D., 1998. "The Syntax of Disfluency in Spontaneous Spoken Language", *HCRC Research Paper HCRC/RP-95*, Edinburgh.

Morgan, N., D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. 2001. The ICSI Meeting Project. In *Proceedings of the Human Language Technology Conference*, San Diego, pp.10-18.

Readings in Corpus Linguistics, ed. G. Sampson and D. McCarthy, London and NY: Continuum International, 2002. Originally circulated on the web in 2000.

Rizzi L. 1982. *Issues in Italian Syntax*, Foris Publications, Dordrecht.

Ronat M. (1982) Logical Form and Prosodic Islands, *Journal of Linguistic Research* 3, 33-48.

Shriberg, E.; R. Bates, and A. Stolcke. A prosody-only decisiontree model for disfluency detection. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proc. EUROSPEECH*, vol. 5, pp. 2383–2386, Rhodes, Greece, 1997.

Shriberg, E.; A. Stolcke, and D. Baron. Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In P. Dalsgaard, B. Lindberg, H. Benner, and Z. Tan, editors, *Proc. EUROSPEECH*, vol. 2, pp. 1359–1362, Aalborg, Denmark, 2001.

Wheatley, B., G. Doddington, C. Hemphill, J. Godfrey, E.C. Holliman, J. McDaniel, and D. Fisher, "Robust Automatic Time Alignment of Orthographic Transcriptions with Unconstrained Speech", *Proc. ICASSP-92*, Vol. I, 533-53

Williams E. (1977) Discourse and Logical Form, *Linguistic Inquiry* 8, MIT Press, Cambridge Mass., 101-139.

Williams E. (1980) Predication, *Linguistic Inquiry* 1, MIT Press, Cambridge Mass., 203-238.