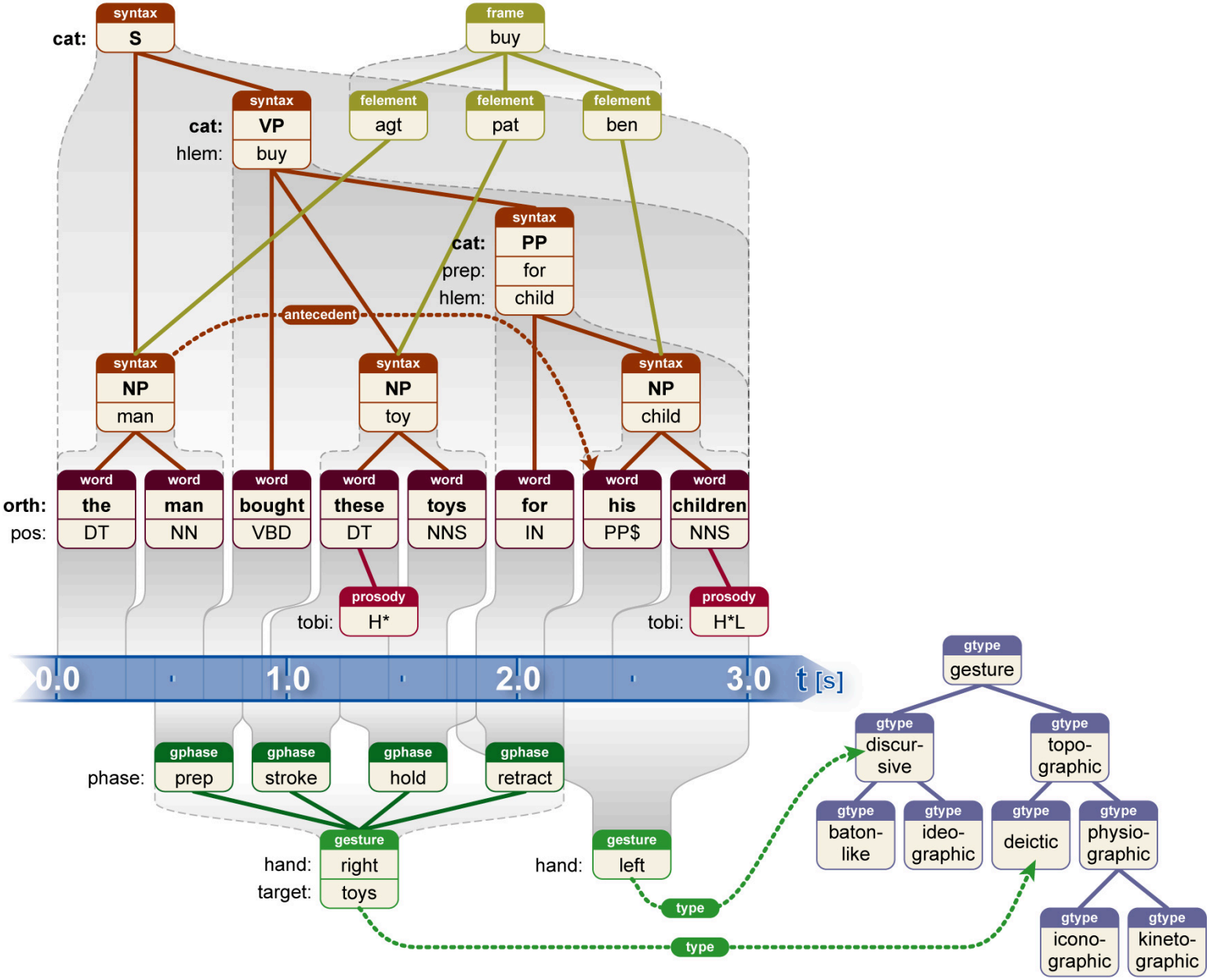
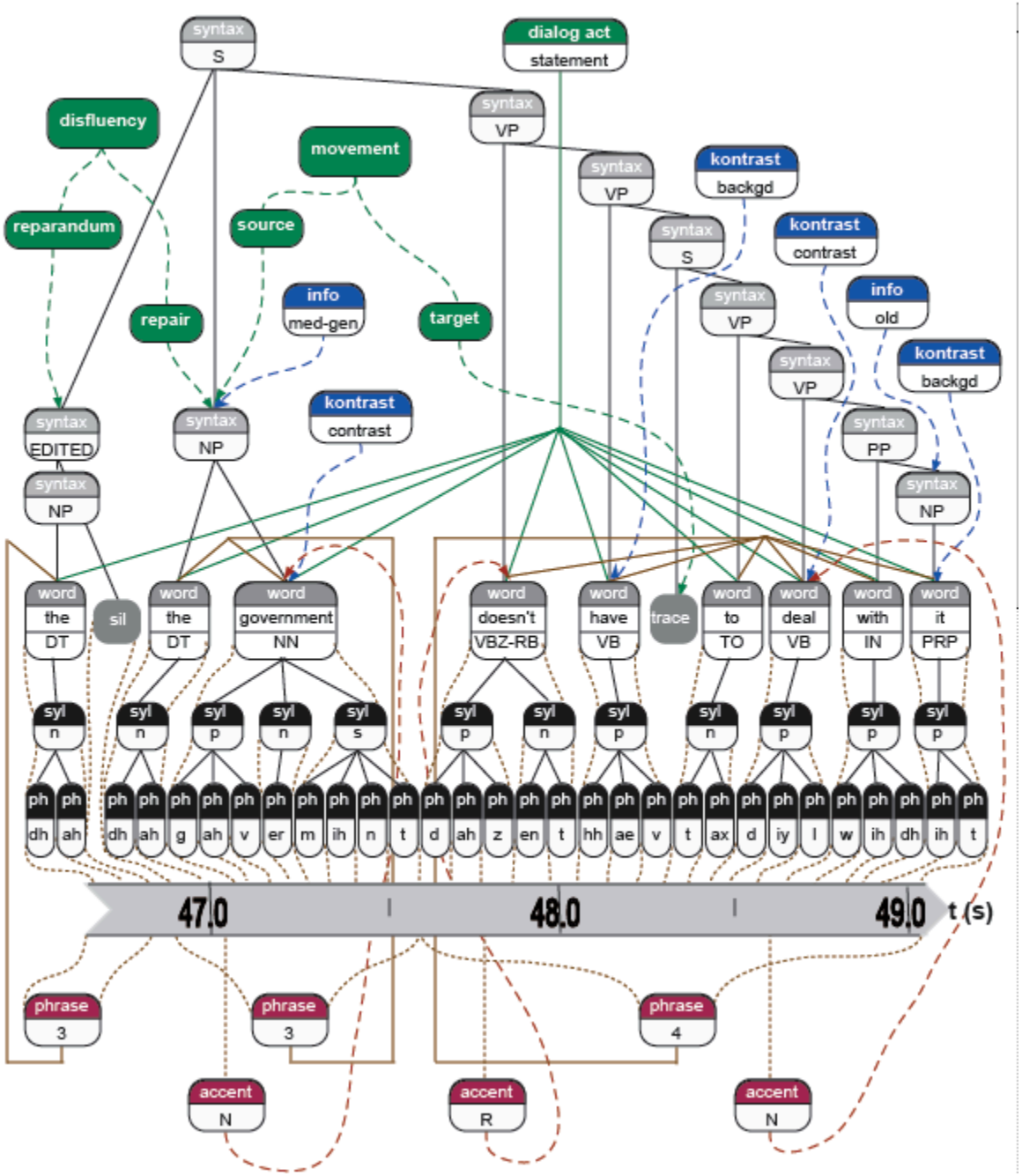
The NITE approach to overlap

Jean Carletta
University of Edinburgh


cat:
syntax
S
frame
buy
cat:
hlem:
syntax
VP
buy
felement
agt
felement
pat
felement
ben
cat:
prep:
hlem:
syntax
PP
for
child
antecedent
syntax
NP
man
syntax
NP
toy
syntax
NP
child
orth:
pos:
word
the
DT
word
man
NN
word
bought
VBD
word
these
DT
word
toys
NNS
word
for
IN
word
his
PP$
word
children
NNS
tobi:
prosody
H*
tobi:
prosody
H*L
0.0
1.0
2.0
3.0
t [s]
phase:
gphase
prep
gphase
stroke
gphase
hold
gphase
retract
hand:
target:
gesture
right
toys
hand:
gesture
left
type
type
gtype
gesture
gtype
discur-sive
gtype
topo-graphic
gtype
baton-like
gtype
ideo-graphic
gtype
deictic
gtype
physio-graphic
gtype
icono-graphic
gtype
kineto-graphic


syntax S
dialog act statement
disfluency
movement
syntax VP
syntax VP
kontrast backgd
reparandum
source
syntax S
kontrast contrast
info med-gen
repair
target
syntax VP
info old
syntax VP
kontrast backgd
syntax EDITED
syntax NP
kontrast contrast
syntax PP
syntax NP
syntax NP
word the DT
sil
word the DT
word government NN
word doesn't VBZ-RB
word have VB
trace
word to TO
word deal VB
word with IN
word it PRP
47.0
48.0
49.0
t (s)
phrase 3
phrase 3
phrase 4
accent N
accent R
accent N

Representation

- Multiple file stand-off annotation
- Out of document references by id using list or ranges of nodes
- XML file structure mirrors major trees in data (one file per colour*speaker)
- “rival” annotations of the same thing can be loaded and compared; builds use a fink-like dependency structure
- Annotation layers can be timed against signals

Ordering

- layers are ordered
- parents draw children in an order that may differ from their layer order
- if two elements do not both participate in any tree, they are unordered with respect to each other
- temporal ordering can also be represented with optional percolation up trees

Multiple file stand-off

extract from o1.A.speech-quality.xml

```
<speechquality nite:id="o1.emphasis.16" type="emphasis">
    <nite:child xlink:href="o1.A.words.xml#xpointer(id('w.2'))" xlink:type="simple" />
</speechquality>
```

extract from o1.A.words.xml

```
<w nite:id="w.2" starttime="356.39" endtime="356.59" c="W">red</w>
```

Processing

- use standard techniques (like XSL) on individual files and constructed trees
  - “knit” and “unknit” for deeper hierarchies
  - “interpose” to get image of alternative parent layer inserted into a hierarchy, where compatible
  - “project” to get image of a layer imposed on a rival of its children, using nearest timing
- special implementation for representing complete data as a graph and traversing or querying it

Very simple example query

($w word)($r reference):

($w@POS = “NN”) && ($r ^ $w)

Return list of 2-tuples of words and referring expressions where the word’s part of speech is NN and the word is in the referring expression.

Metadata file

Like set of DTDs for the XML files plus:

- connections between the files
- list of "observations" (coded dialogues/ group discussions/texts)
- catalog for finding signals and data on disk

Validation

- generate one schema from the metadata file
- can be used to validate any individual file or any knitted tree, including what had been represented as links
- can’t represent/validate cross-hierarchy constraints (but can often query for violations)
- content model for layers assumed to be

(t1 t2 ... tn)*

Advantages

- id-based stand-off fairly robust when base transcription changes
- individual files and constructed trees amenable to standard XML processing
- distributed simultaneous authorship of annotations
- easy to load subsets of the tags; query stays the same

Disadvantages

- lots of files
- in links, files identified by name only, with location separated off in metadata
- standard mechanisms that deal with XLink/xpointer are slow on links