Incrementality, Alignment and Split Utterances

Matthew Purver
with Ruth Kempson, Pat Healey, Eleni Gregoromichelaki,
Christine Howes, Wilfried Meyer-Viol, Graham White

The Dynamics of Conversational Dialogue (DynDial)
ESRC-RES-062-23-0962
www.kcl.ac.uk/research/groups/ds

March 11, 2010

Outline

1. Dialogue and Incrementality
    - Split Utterances and Alignment
2. Dynamic Syntax (DS)
    - A Quick Introduction to DS
    - DS and Dialogue Modelling
3. Empirical Investigations
    - Priming - Corpus Study
    - Split Utterances - Corpus Study
    - Split Utterances - Experiments
4. Dynamic Syntax & Type Theory with Records (TTR)
    - Adding TTR to DS
    - Fragments & Split Utterances in DS/TTR

Outline

1. Dialogue and Incrementality
    - Split Utterances and Alignment
2. Dynamic Syntax (DS)
    - A Quick Introduction to DS
    - DS and Dialogue Modelling
3. Empirical Investigations
    - Priming - Corpus Study
    - Split Utterances - Corpus Study
    - Split Utterances - Experiments
4. Dynamic Syntax & Type Theory with Records (TTR)
    - Adding TTR to DS
    - Fragments & Split Utterances in DS/TTR

Dialogue and Incrementality

- Plenty of interest in dialogue
  - Formal models of dialogue moves, IS update, fragments
- Plenty of interest in incrementality
  - Incremental processing in psycholinguistics
  - Incremental parsing and generation in computational linguistics
- Increasing interest in incrementality in dialogue
  - e.g. [Schlangen and Skantze, 2009, Schuler et al., 2009]
  - Speeding up dialogue systems
  - Processing human-human dialogue
  - People do it this way . . .

The Dynamics of Conversational Dialogue

- An ESRC project, joint between QMUL and KCL
  - formal/computational linguists, logicians, experimental psychologists
- Linguistic modelling using Dynamic Syntax [Kempson et al., 2001]
  - inherently incremental grammar formalism
- Empirical studies using corpora and experiments

The Dynamics of Conversational Dialogue

- An ESRC project, joint between QMUL and KCL
    - formal/computational linguists, logicians, experimental psychologists
- Linguistic modelling using Dynamic Syntax [Kempson et al., 2001]
    - inherently incremental grammar formalism
- Empirical studies using corpora and experiments
    - Non-sentential utterances
    - Clarification requests
    - Split utterances
    - Priming/alignment

Split Utterances

- Utterances containing a change in speaker
  - . . . and therefore a change in hearer

Split Utterances

- Utterances containing a change in speaker
  - . . . and therefore a change in hearer

A: The profit for the group is 190,000.
B: Which is superb. (“expansion”)

Split Utterances

- Utterances containing a change in speaker
  - …and therefore a change in hearer

A: The profit for the group is 190,000.
B: Which is superb. ("expansion")

A: Before that then if they were ill
G: They get nothing. ("completion")

Split Utterances

- Utterances containing a change in speaker
  - . . . and therefore a change in hearer

A: The profit for the group is 190,000.
B: Which is superb. ("expansion")

A: Before that then if they were ill
G: They get nothing. ("completion")

- Fundamental requirement for incremental processing
  - A good test for syntactic and semantic dependencies
- Treatment for one particular kind [Poesio and Rieser, 2010]
  - LTAG grammar and conversational-event-based plan recognition

Split Utterances

- Particularly interesting from an incrementality point of view

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle,

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle, to go to the farm shop

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle, to go to the farm shop

(2) Hugh: Ruth visited Trecastle, to go to the
Alex: farm shop

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle, to go to the farm shop

(2) Hugh: Ruth visited Trecastle, to go to the
Alex: farm shop

- Splits can occur across syntactic/semantic dependencies:

(3) A: Have you read ...
B: any of your chapters? Not yet.

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle, to go to the farm shop

(2) Hugh: Ruth visited Trecastle, to go to the
Alex: farm shop

- Splits can occur across syntactic/semantic dependencies:

(3) A: Have you read ...
B: any of your chapters? Not yet.

- Not just a case of splitting a *string*

Split Utterances

- Particularly interesting from an incrementality point of view
- Where can splits occur? Within constituents?

(1) Hugh: Ruth visited
Alex: Trecastle, to go to the farm shop

(2) Hugh: Ruth visited Trecastle, to go to the
Alex: farm shop

- Splits can occur across syntactic/semantic dependencies:

(3) A: Have you read ...
B: any of your chapters? Not yet.

- Not just a case of splitting a *string*
- How common are these really?
- Where do splits really occur (how incremental must we be)?

Priming and/or Alignment

- Tendency to repeat previously used material
    - words
    - syntactic structures [Branigan et al., 2000]
    - multi-word expressions
    - ways of referring [Garrod and Anderson, 1987]
- Both self- and other- effects [Pickering and Ferreira, 2008]
- Interesting for models of incremental processing
    - (. . . especially in the case of split utterances . . .)
    - what phenomena are primed/aligned? (and therefore represented)?
    - evidence for independence of lexicon/syntax/semantics?
- Most data from controlled experimental settings
- What does this tell us about real dialogue?

Outline

1. Dialogue and Incrementality
   - Split Utterances and Alignment
2. Dynamic Syntax (DS)
   - A Quick Introduction to DS
   - DS and Dialogue Modelling
3. Empirical Investigations
   - Priming - Corpus Study
   - Split Utterances - Corpus Study
   - Split Utterances - Experiments
4. Dynamic Syntax & Type Theory with Records (TTR)
   - Adding TTR to DS
   - Fragments & Split Utterances in DS/TTR

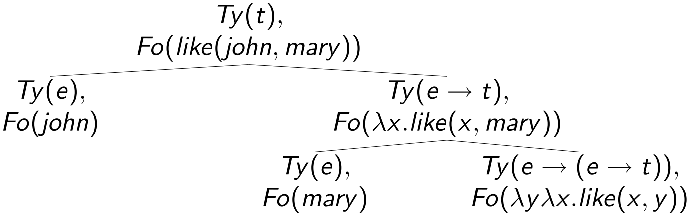
Dynamic Syntax

- An inherently incremental grammatical framework
- Word-by-word incremental construction of semantic interpretation:
    - no autonomous level of syntax
    - “syntax” defined via constraints on incremental semantic structure-building
    - “grammar” is a set of procedures for incremental parsing
    - “trees” are semantic representations defined using LoFT [Blackburn and Meyer-Viol, 1994]
- Monotonic growth with underspecification-plus-enrichment
- Procedural definitions: constraints on *how* interpretations are built

DS Trees as semantic representations

- End product of parsing is a semantic tree
- Nodes decorated with $Ty()$ type and $Fo()$ formula labels

“John likes Mary”:



$Ty(t)$, $Fo(like(john, mary))$

$Ty(e)$, $Fo(john)$

$Ty(e \rightarrow t)$, $Fo(\lambda x.like(x, mary))$

$Ty(e)$, $Fo(mary)$

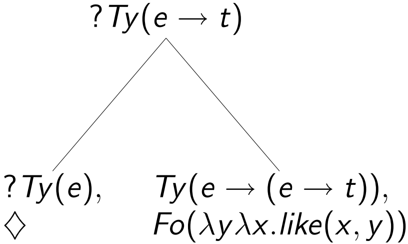
$Ty(e \rightarrow (e \rightarrow t))$, $Fo(\lambda y \lambda x.like(x, y))$

- Daughter order does not reflect sentence order!
- Nodes interpretable as terms in the λ-calculus
- NPs map onto terms of type e using the ϵ-calculus.

Actions as tree-building procedures

- Incremental tree growth driven by *requirements* e.g. $?Ty(t)$
- Node under development marked by *pointer* $\diamondsuit$
- Words induce sets of *lexical* actions: *"like"*

IF $?Ty(e \rightarrow t)$
THEN make($\langle\downarrow_1\rangle$);go($\langle\downarrow_1\rangle$);
put($Fo(\lambda y \lambda x.like(x,y))$);
put($Ty(e \rightarrow (e \rightarrow t))$)
go($\langle\uparrow_1\rangle$); make($\langle\downarrow_0\rangle$);
go($\langle\downarrow_0\rangle$); put($?Ty(e)$)
ELSE ABORT



$?Ty(e \rightarrow t)$

$?Ty(e)$, $\diamondsuit$

$Ty(e \rightarrow (e \rightarrow t))$, $Fo(\lambda y \lambda x.like(x,y))$

- General *computational* actions are also available e.g. requirement fulfillment, beta-reduction

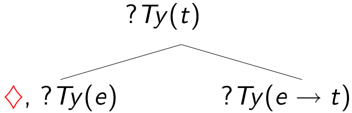
Unfolding then building up the tree

Processing *Someone fainted*

$?Ty(t)$, $\diamondsuit$

Unfolding then building up the tree

Processing *Someone fainted*



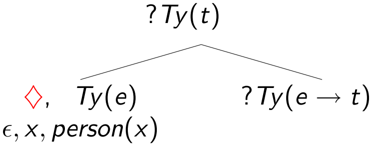
$?Ty(t)$

$\diamondsuit, ?Ty(e)$

$?Ty(e \rightarrow t)$

Unfolding then building up the tree

Processing *Someone fainted*



$?Ty(t)$

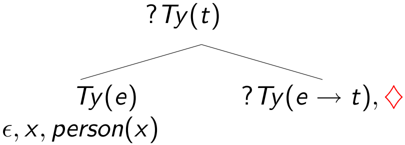
$\diamondsuit, \quad Ty(e)$

$\epsilon, x, person(x)$

$?Ty(e \rightarrow t)$

Unfolding then building up the tree

Processing *Someone fainted*



$?Ty(t)$

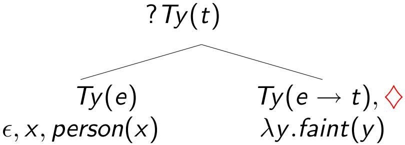
$Ty(e)$

$\epsilon, x, person(x)$

$?Ty(e \rightarrow t), \Diamond$

Unfolding then building up the tree

Processing *Someone fainted*



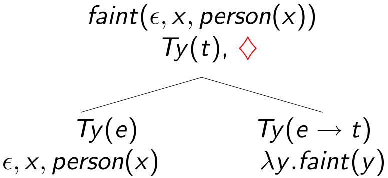
$?Ty(t)$

$Ty(e)$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t), \diamondsuit$
$\lambda y.faint(y)$

Unfolding then building up the tree

Processing *Someone fainted*



$faint(\epsilon, x, person(x))$
$Ty(t), \diamondsuit$

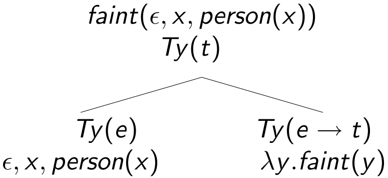
$Ty(e)$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

Unfolding then building up the tree

Processing *Someone fainted*

$\rightsquigarrow faint(\epsilon, x, person(x))$



$faint(\epsilon, x, person(x))$
$Ty(t)$

$Ty(e)$
$\epsilon, x, person(x)$

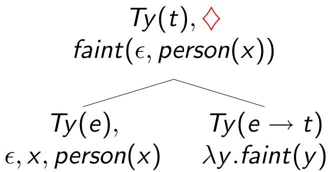
$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

* Generating *Someone fainted*

GOAL TREE



$Ty(t), \diamondsuit$
$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

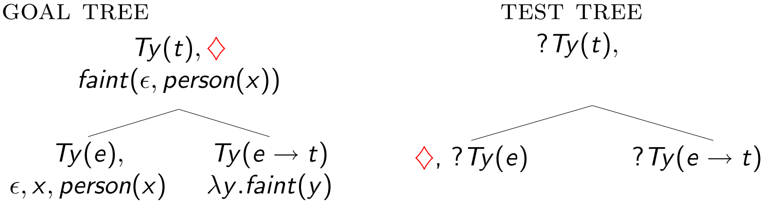
TEST TREE

$?Ty(t), \diamondsuit$

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

* Generating *Someone fainted*



GOAL TREE

$Ty(t), \diamondsuit$
$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

TEST TREE

$?Ty(t),$

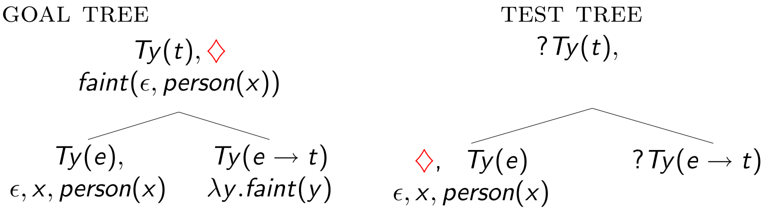
$\diamondsuit, ?Ty(e)$

$?Ty(e \rightarrow t)$

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

* Generating *Someone fainted*



GOAL TREE

$Ty(t), \diamondsuit$
$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

TEST TREE

$?Ty(t),$

$\diamondsuit, \quad Ty(e)$
$\epsilon, x, person(x)$

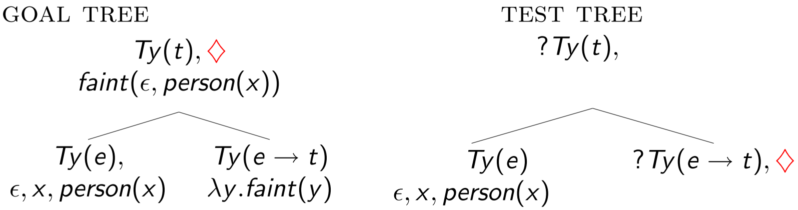
$?Ty(e \rightarrow t)$

Gen: “Someone

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

* Generating *Someone fainted*



GOAL TREE

$Ty(t), \diamondsuit$

$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

TEST TREE

$?Ty(t),$

$Ty(e)$
$\epsilon, x, person(x)$

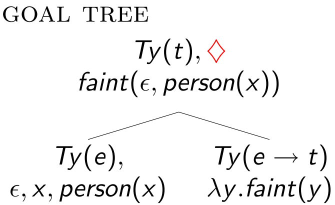
$?Ty(e \rightarrow t), \diamondsuit$

Gen: “Someone

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

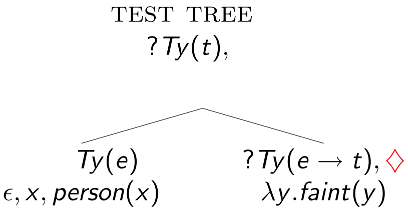
* Generating *Someone fainted*


GOAL TREE

$Ty(t), \diamondsuit$
$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$


TEST TREE

$?Ty(t),$

$Ty(e)$
$\epsilon, x, person(x)$

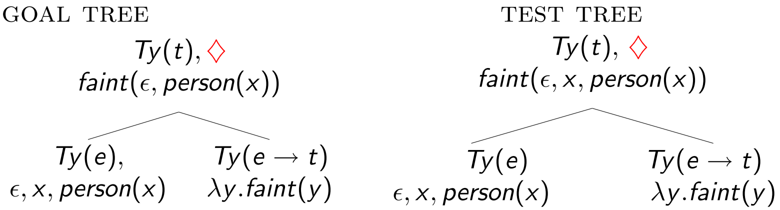
$?Ty(e \rightarrow t), \diamondsuit$
$\lambda y.faint(y)$

Gen: “Someone fainted”

Generation

- Speakers go through the same tree-growth actions, except they also have a somewhat richer goal tree.
- Each word licensed must update partial tree towards the goal tree via *subsumption* constraint

* Generating *Someone fainted*



GOAL TREE

$Ty(t), \diamondsuit$
$faint(\epsilon, person(x))$

$Ty(e),$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

TEST TREE

$Ty(t), \diamondsuit$
$faint(\epsilon, x, person(x))$

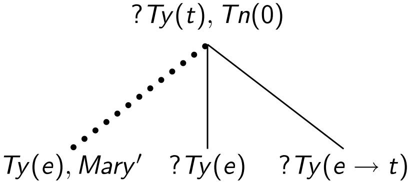
$Ty(e)$
$\epsilon, x, person(x)$

$Ty(e \rightarrow t)$
$\lambda y.faint(y)$

Gen: "Someone fainted"

Underspecification: structural

- “Unfixed” nodes - building underspecified tree relations



$?Ty(t), Tn(0)$

$Ty(e), Mary'$ $\quad ?Ty(e)$ $\quad ?Ty(e \rightarrow t)$

- Left-dislocation “Mary, John likes”

Underspecification: content

- Pronouns project META-VARIABLES (**U**)
- Substituted by item from context during construction

Underspecification: content

- Pronouns project META-VARIABLES (**U**)
- Substituted by item from context during construction

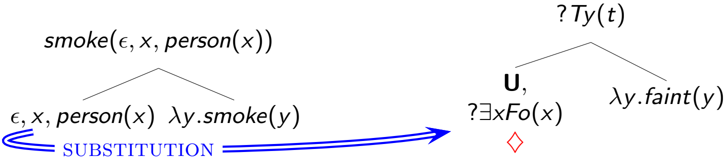
(1) Someone smoked He fainted.

Underspecification: content

- Pronouns project META-VARIABLES (**U**)
- Substituted by item from context during construction

(1) Someone smoked He fainted.

TREE AS CONTEXT: TREE UNDER CONSTRUCTION:



$smoke(\epsilon, x, person(x))$

$\epsilon, x, person(x)$ $\lambda y.smoke(y)$

SUBSTITUTION

$?Ty(t)$

$\mathbf{U}$, $?\exists x Fo(x)$ ◇

$\lambda y.faint(y)$

Underspecification: ellipsis

- Auxiliaries also project META-VARIABLES (**V**)

  Substituted by item from context in the same way

Underspecification: ellipsis

- Auxiliaries also project META-VARIABLES (**V**)

  Substituted by item from context in the same way

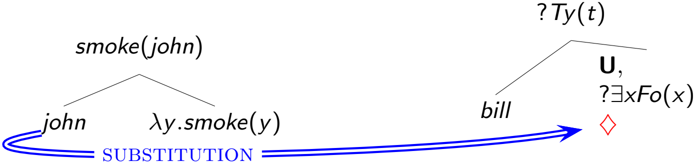
(1) John smoked Bill did too.

Underspecification: ellipsis

- Auxiliaries also project META-VARIABLES (**V**)
  Substituted by item from context in the same way

(1) John smoked Bill did too.

TREE AS CONTEXT: TREE UNDER CONSTRUCTION:



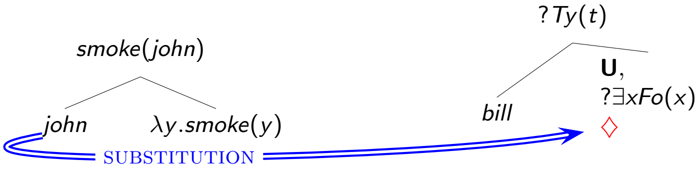
Underspecification: ellipsis

- Auxiliaries also project META-VARIABLES (**V**)

  Substituted by item from context in the same way

(1) John smoked Bill did too.

TREE AS CONTEXT: TREE UNDER CONSTRUCTION:

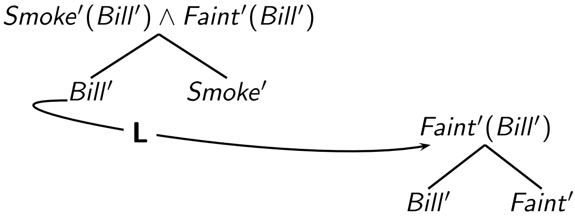


- Alternatively can use *actions* from context (sloppy readings)
- Simple model of *context* containing previous (partial) trees and action sequences

Context-dependence: LINKed tree-pairs

- **Relative clauses**: pairs of LINKed trees evaluated as conjunction

  e.g. Bill, who fainted, smokes.



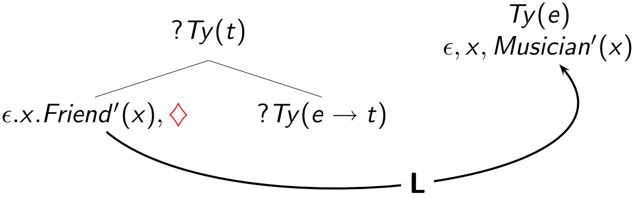
Appositions as LINKed trees

e.g. A friend, a musician, smokes.

Appositions as LINKed trees

e.g. A friend, a musician, smokes.

- Partial tree as context with term enriched by LINKed tree of same type
- Parsing *A friend, a musician*



$?Ty(t)$

$\epsilon.x.Friend'(x), \Diamond$

$?Ty(e \rightarrow t)$

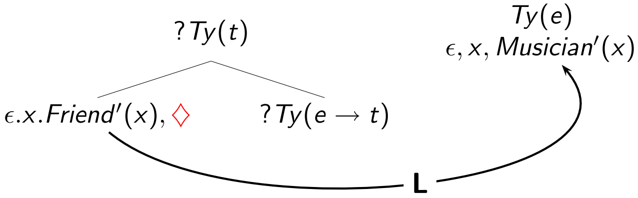
$Ty(e)$

$\epsilon, x, Musician'(x)$

Appositions as LINKed trees

e.g. A friend, a musician, smokes.

- Partial tree as context with term enriched by LINKed tree of same type
- Parsing *A friend, a musician*



$?Ty(t)$

$\epsilon.x.Friend'(x), \Diamond$ $\quad ?Ty(e \rightarrow t)$

$Ty(e)$
$\epsilon, x, Musician'(x)$

L

Evaluation of LINKed nodes both of type e yields composite term:

$\epsilon, x, Friend'(x) \wedge Musician'(x)$

Final formula: $Smoke'(\quad \epsilon, x, Friend'(x) \wedge Musician'(x) \quad)$

DS and Split Utterances

- DS *seems* well suited for split utterances
- Inherent word-by-word incrementality
- Well-defined partial structures at each point
- Same actions and partial structures in parsing and generation
- Grammatical constraints via semantics rather than "syntax"
    - Not licensing a string
    - Splits should be possible anywhere

DS and Split Utterances

- DS *seems* well suited for split utterances
- Inherent word-by-word incrementality
- Well-defined partial structures at each point
- Same actions and partial structures in parsing and generation
- Grammatical constraints via semantics rather than "syntax"
    - Not licensing a string
    - Splits should be possible anywhere

$?Ty(t)$

$john$ $\quad ?Ty(e \rightarrow t), \Diamond$

$smoke(john), \Diamond$

$john$ $\quad \lambda y.smoke(y)$

DS and Split Utterances

- DS *seems* well suited for split utterances
- Inherent word-by-word incrementality
- Well-defined partial structures at each point
- Same actions and partial structures in parsing and generation
- Grammatical constraints via semantics rather than "syntax"
    - Not licensing a string
    - Splits should be possible anywhere
- Is it too general (what are the real constraints)?
- Is it too simplistic (what do split utterances *mean*)?

DS and Priming/Alignment

- DS *seems* well suited to explain priming/alignment phenomena
- Use of actions at all levels of processing
- Availability of recent action (sequences) for re-use
    - Lexical choice and disambiguation
    - Syntactic phenomena (e.g. DO/PO alternation [Branigan et al., 2000])
    - Semantic/pragmatic phenomena (e.g. routines [Garrod and Anderson, 1987], ellipsis construal [Hardt, 2008])

DS and Priming/Alignment

- DS *seems* well suited to explain priming/alignment phenomena
- Use of actions at all levels of processing
- Availability of recent action (sequences) for re-use
    - Lexical choice and disambiguation
    - Syntactic phenomena (e.g. DO/PO alternation [Branigan et al., 2000])
    - Semantic/pragmatic phenomena (e.g. routines [Garrod and Anderson, 1987], ellipsis construal [Hardt, 2008])

```
IF      $?Ty(e \rightarrow t)$
THEN    make($\langle\downarrow_1\rangle$);go($\langle\downarrow_1\rangle$);
        put($Fo(\lambda y \lambda x.like(x, y))$);
        put($Ty(e \rightarrow (e \rightarrow t))$)
        go($\langle\uparrow_1\rangle$); make($\langle\downarrow_0\rangle$);
        go($\langle\downarrow_0\rangle$); put($?Ty(e)$)
ELSE    ABORT
```

DS and Priming/Alignment

- DS *seems* well suited to explain priming/alignment phenomena
- Use of actions at all levels of processing
- Availability of recent action (sequences) for re-use
    - Lexical choice and disambiguation
    - Syntactic phenomena (e.g. DO/PO alternation [Branigan et al., 2000])
    - Semantic/pragmatic phenomena (e.g. routines [Garrod and Anderson, 1987], ellipsis construal [Hardt, 2008])
- Does this really explain general (non-lexical) effects?
    - Branigan et al found weaker cross-verb effects
    - (re-use of *computational* action sequences?)
- Do we see them in real dialogue?

Outline

1. Dialogue and Incrementality
   - Split Utterances and Alignment
2. Dynamic Syntax (DS)
   - A Quick Introduction to DS
   - DS and Dialogue Modelling
3. Empirical Investigations
   - Priming - Corpus Study
   - Split Utterances - Corpus Study
   - Split Utterances - Experiments
4. Dynamic Syntax & Type Theory with Records (TTR)
   - Adding TTR to DS
   - Fragments & Split Utterances in DS/TTR

Empirical Investigations

- What do these phenomena really look like?

Empirical Investigations

- What do these phenomena really look like?
- What's the deal with lexical and syntactic priming?
    - Do we see them in ordinary dialogue?
    - Can we tell which effect is greater?

Empirical Investigations

- What do these phenomena really look like?
- What's the deal with lexical and syntactic priming?
    - Do we see them in ordinary dialogue?
    - Can we tell which effect is greater?
- Do split utterances really behave the way we think?
    - How common are they?
    - Where does the split happen?
    - What do they mean?

Priming: Designing a corpus experiment

- DS seems to predict lexical(-syntactic) effects more than general syntactic effects
- Previous dialogue experiments (e.g. [Reitter et al., 2006]) suggest that:
    - general syntactic effects are stronger in task-specific dialogue than in general conversation
    - general syntactic effects are stronger within-person than cross-person
- But no direct control condition:
    - what about dialogue structure effects?
    - how similar would recent turns be by chance?
    - Switchboard corpus is strange

Corpus experiment: Method

- DCPSE corpus, all 2-person dialogues from 3 largest genre samples:
  - face-to-face formal (60 dialogues, 90,000 words)
  - face-to-face informal (91 dialogues, 403,000 words)
  - telephone conversations (89 dialogues, 77,000 words)
- For each dialogue D, create a “fake” control dialogue:
  - keep all turns from first speaker $S1_D$
  - choose a different dialogue D', matching by length and within genre
  - interleave the turns from $S1_D$ with those from $S2_{D'}$
- Compare average turn similarity between real and control dialogues

Corpus experiment: Method

A:	Hello	A':	Hi
B:	Hi	B':	Hello
A:	How are you?	A':	What's up?
B:	Fine - you?	B':	Not much
A:	Yeah fine thanks	A':	Me neither
B:	Uh-huh	B':	Uh-huh

Corpus experiment: Method

A: Hello

B': Hello

A: How are you?

B': Not much

A: Yeah fine thanks

B': Uh-huh

Corpus experiment: Method

A:	Hello
B':	Hello
A:	How are you?
B':	Not much
A:	Yeah fine thanks
B':	Uh-huh

Corpus experiment: Method

A: Hello
B': Uh-huh
A: How are you?
B': Hello
A: Yeah fine thanks
B': Not much

Corpus experiment: Lexical results

- Lexical similarity expressed via word pair kernel:
    - number of matching word pairs between turns A and $B = N_{AB}$
    - similarity $S_{lex} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$

Corpus experiment: Lexical results

- Lexical similarity expressed via word pair kernel:
  - number of matching word pairs between turns A and $B = N_{AB}$
  - similarity $S_{lex} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$
- Real dialogues mean other-person similarity

  $S_{lex} = 0.094\ (SD = 0.04)$
- Control dialogues mean other-person similarity

  $S_{lex} = 0.059\ (SD = 0.03)$

Corpus experiment: Lexical results

- Lexical similarity expressed via word pair kernel:
    - number of matching word pairs between turns A and $B = N_{AB}$
    - similarity $S_{lex} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$
- Real dialogues mean other-person similarity

  $S_{lex} = 0.094 \ (SD = 0.04)$
- Control dialogues mean other-person similarity

  $S_{lex} = 0.059 \ (SD = 0.03)$
- ANOVA for real vs. control shows difference is reliable:

  $F_{(1,253)} = 106.55, \ p = 0.00$

Corpus experiment: Syntactic results (1)

- Syntactic similarity via tree kernel (variant of [Moschitti, 2006]):
  - number of matching non-terminal syntactic rule pairs between turns A and $B = N_{AB}$
  - similarity $S_{syn} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$

Corpus experiment: Syntactic results (1)

- Syntactic similarity via tree kernel (variant of [Moschitti, 2006]):
  - number of matching non-terminal syntactic rule pairs between turns A and $B = N_{AB}$
  - similarity $S_{syn} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$
- Real dialogues mean other-person similarity

  $S_{syn} = 0.19$ $(SD = 0.06)$
- Control dialogues mean other-person similarity

  $S_{syn} = 0.18$ $(SD = 0.06)$

Corpus experiment: Syntactic results (1)

- Syntactic similarity via tree kernel (variant of [Moschitti, 2006]):
    - number of matching non-terminal syntactic rule pairs between turns A and $B = N_{AB}$
    - similarity $S_{syn} = \frac{N_{AB}}{\sqrt{N_{AA}.N_{BB}}}$
- Real dialogues mean other-person similarity
  $S_{syn} = 0.19$ $(SD = 0.06)$
- Control dialogues mean other-person similarity
  $S_{syn} = 0.18$ $(SD = 0.06)$
- ANOVA for real vs. control shows difference *not* reliable:
  $F_{(1,253)} = 1.32, \;\; p = 0.25$
- But: a reliable effect of genre ($F_{(2,237)} = 20.13, p = 0.00$):

	formal	informal	telephone
real	0.21	0.19	0.17
control	0.21	0.18	0.16

Corpus experiment: Syntactic results (2)

- What's the influence of lexical similarity on syntactic similarity?
- Linear Mixed Model analysis can tell us:
  - subject, dialogue as random factors
  - real/control type as fixed factor
  - lexical similarity as covariate

Corpus experiment: Syntactic results (2)

- What's the influence of lexical similarity on syntactic similarity?
- Linear Mixed Model analysis can tell us:
    - subject, dialogue as random factors
    - real/control type as fixed factor
    - lexical similarity as covariate
- Parameter estimate for S_{syn} negative
- Marginal ("corrected") means: $S_{syn} = 0.184$ real, $S_{syn} = 0.211$ control
- Reliable difference: $p = 0.01$
- i.e. S_{syn} is *lower* than chance when S_{lex} taken into account

Corpus experiment: Syntactic results (2)

- What's the influence of lexical similarity on syntactic similarity?
- Linear Mixed Model analysis can tell us:
    - subject, dialogue as random factors
    - real/control type as fixed factor
    - lexical similarity as covariate
- Parameter estimate for S_{syn} negative
- Marginal ("corrected") means:
  $S_{syn} = 0.184$ real, $S_{syn} = 0.211$ control
- Reliable difference: $p = 0.01$
- i.e. S_{syn} is *lower* than chance when S_{lex} taken into account
- Checked on BNC spoken portion (bigger but not parsed)
    - parsed using C&C CCG parser, Stanford CFG parser
    - results the same

Corpus experiment: Conclusions

- We can measure the effect of lexical priming
- We can't measure the effect of syntactic priming
  - It appears to be negative when lexical effect taken into account
  - Even if it exists, it must be small (relative to the lexical effect)
- We can measure the effect of genre on syntactic similarity
  - This seems to agree with (some of) [Reitter et al., 2006]'s results

Corpus experiment: Conclusions

- We can measure the effect of lexical priming
- We can't measure the effect of syntactic priming
  - It appears to be negative when lexical effect taken into account
  - Even if it exists, it must be small (relative to the lexical effect)
- We can measure the effect of genre on syntactic similarity
  - This seems to agree with (some of) [Reitter et al., 2006]'s results
- A grammar which associates syntactic effects with lexical entries might be on the right track . . .
- We'd like to know more about individual phenomena . . .

Split Utterances: Corpus Study

- Take a portion of the BNC (as annotated by [Fernández, 2006])
- Find all the split utterances
    - not just other-person cases [Skuplik, 1999, Szczepek, 2000]
    - or particular CA phenomena [Lerner, 2004, Rühlemann, 2007]
- See how often they occur, for same- and other-person cases
- See how variable the split point is
    - Completeness/constituency of the two halves *completion/expansion*
    - Dependencies across the split
- See what happens in between . . .

Corpus Study: Annotation Schema

- A1: I'll definitely use that

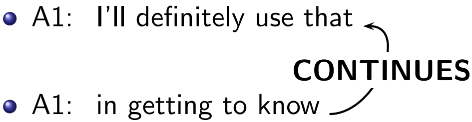
Corpus Study: Annotation Schema

- A1: I'll definitely use that ⟵ **END-COMPLETE=Y** —

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- A1: in getting to know

Corpus Study: Annotation Schema

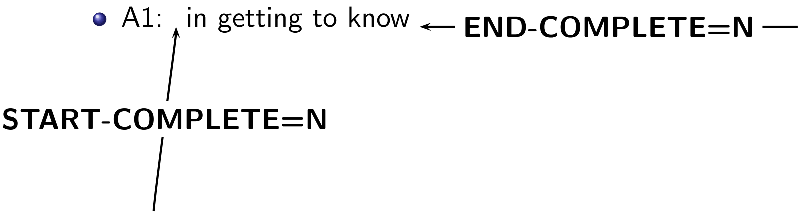


- A1: I'll definitely use that

CONTINUES

- A1: in getting to know

Corpus Study: Annotation Schema



- A1: I'll definitely use that
- A1: in getting to know ← **END-COMPLETE=N** —

START-COMPLETE=N

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- A1: in getting to know
- A1: new year seven

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- A1: in getting to know
- A1: new year seven

CONTINUES

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- A1: in getting to know
- A1: new year seven ⟵ **END-COMPLETE=Y** —

START-COMPLETE=N

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- UX: *[reading]* Get a headache?
- A1: in getting to know

- A1: new year seven

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- UX: *[reading]* Get a headache?
- A1: [in getting to know]
- A2: [Year seven]
- A1: new year seven

Corpus Study: Annotation Schema

- A1: I'll definitely use that
- UX: *[reading]* Get a headache?
- A1: [in getting to know]
- A2: [Year seven]
- A1: new [year seven]
- A2: [Oh yeah] for year seven

Corpus Study: Observations

- They're common: 19% of all contributions continue something
- 85% of these are same-person cases
- 15% are other-person cases
    - this is about 3% of all dialogue contributions (i.e. about as common as clarification)

Corpus Study: Observations

- They're common: 19% of all contributions continue something
- 85% of these are same-person cases
- 15% are other-person cases
    - this is about 3% of all dialogue contributions (i.e. about as common as clarification)
- Many are within-turn (although these are still interesting!)
- Some may be artefacts of the BNC transcription protocol
    - overlapping speech forces a split into two contributions
- But even without all these, 10% of contributions are SUs

Corpus Study: Observations

- The first part is often (but not always) incomplete: 26-28% of cases
- Some neat "syntactic" categories exist, as expected
- But these only cover 50-60% of cases
- Splits can apparently happen at any syntactic point, including inside NPs/PPs:

  (1) F: We are going to call you the
  U: Wallering

  (2) A: And they went over just to be fitted with the
  G: just fitted with the brass

- Note the presence of repair: only 5% of cases

Corpus Study: Observations

- They're not always adjacent:
  - Same-person: 35% separated by a backchannel, 20% by 1 or more other turns
  - Other-person: 5% separated by a backchannel, 5% by 1 or more other turns
- Intervening material is often a clarification:

(3) J: If you press N
S: N?
J: N for name, it'll let you type in the docu- document name.

- The antecedent for clarification is often incomplete . . .
  - (hard to establish propositional content/intention of antecedent)

Corpus Study: Observations

- Continuations often don't perform the same *function* as the antecedent:

(4) G: Had their own men
A: unload the boats?
G: unload the boats, yes.

(5) J: How does it generate?
M: It's generated with a handle and
J: Wound round?
M: Yes, wind them round

- Very often a clarification request, but others possible e.g. confirmation, reformulation
- Not quite as simple as just completing a semantic structure . . .

Corpus Study: Conclusions

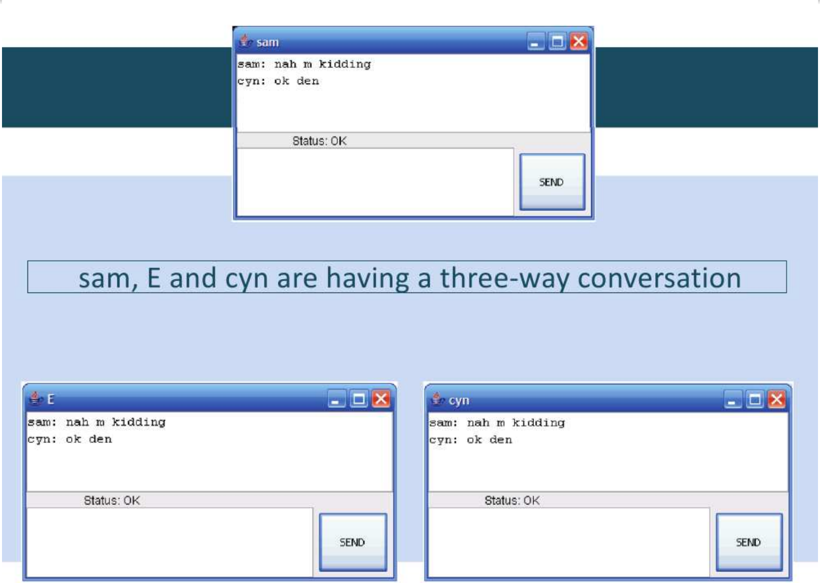
- Some conclusions play right into DS's hands ...
  - Splits happen within syntactic/semantic "constituents"
  - Not always collaborative as per [Poesio and Rieser, 2010]
  - Intervening turns use incomplete antecedents (partial trees)

Corpus Study: Conclusions

- Some conclusions play right into DS's hands . . .
  - Splits happen within syntactic/semantic "constituents"
  - Not always collaborative as per [Poesio and Rieser, 2010]
  - Intervening turns use incomplete antecedents (partial trees)
- . . . but some don't:
  - Repair
  - Clarifications

Experimental Study: the DiET chattool

- Corpora tell us nothing about processing questions
    - SUs may be common, but are they easy/hard to process?
- DiET: a toolbox for experimenting with dialogue [Healey et al., 2003]
- Basic setup: a multi-way chat tool, a bit like MSN Messenger
- Communication is mediated by a server, allowing controlled manipulations
    - transform real turns
    - introduce "fake" turns
- Use this to introduce split utterances, and observe the effects

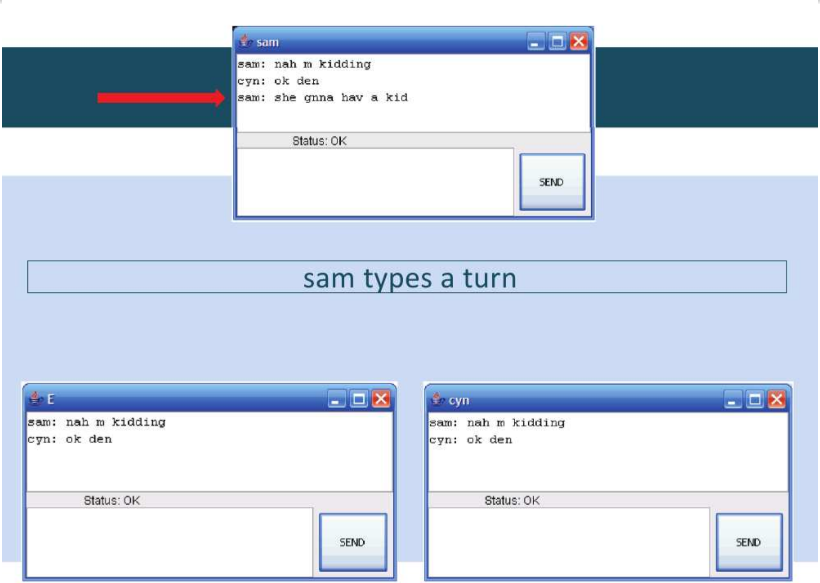


```
sam: nah m kidding
cyn: ok den
```

sam, E and cyn are having a three-way conversation

```
sam: nah m kidding
cyn: ok den
```

```
sam: nah m kidding
cyn: ok den
```

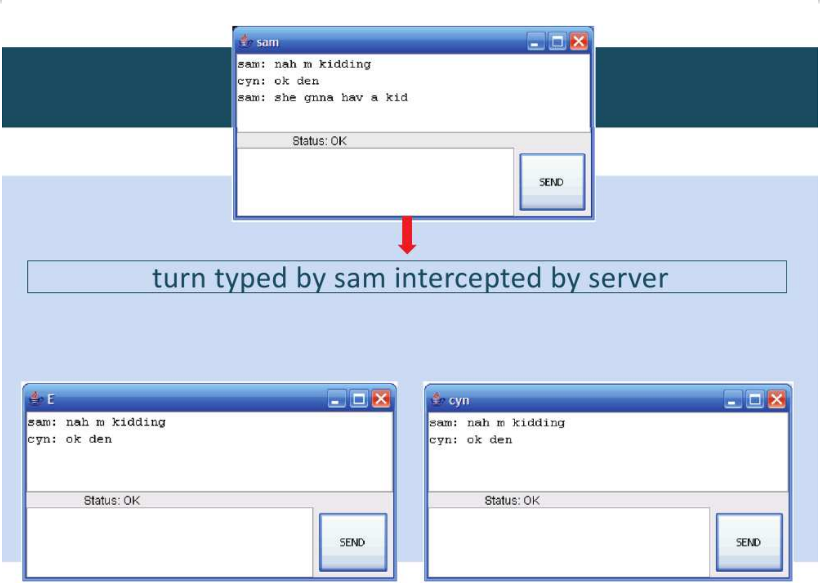


sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid

sam types a turn

sam: nah m kidding
cyn: ok den

sam: nah m kidding
cyn: ok den

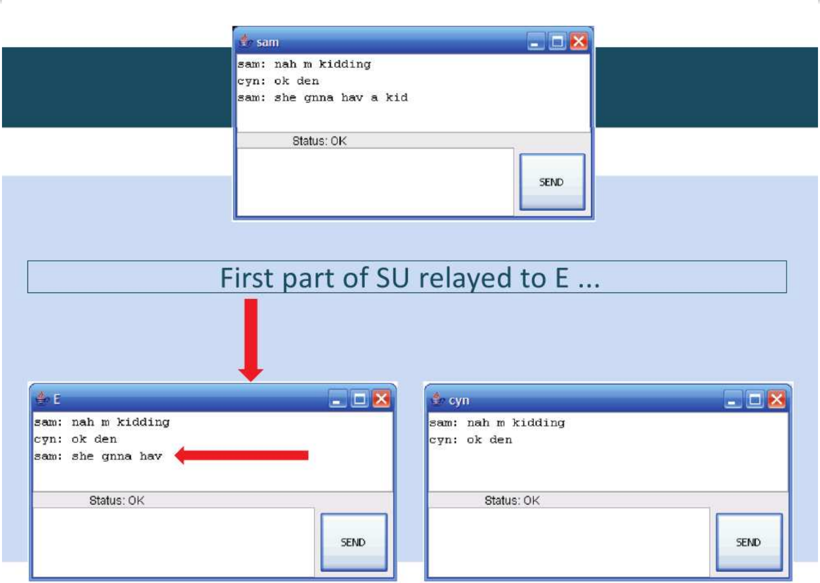


```
sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid
```

turn typed by sam intercepted by server

```
sam: nah m kidding
cyn: ok den
```

```
sam: nah m kidding
cyn: ok den
```

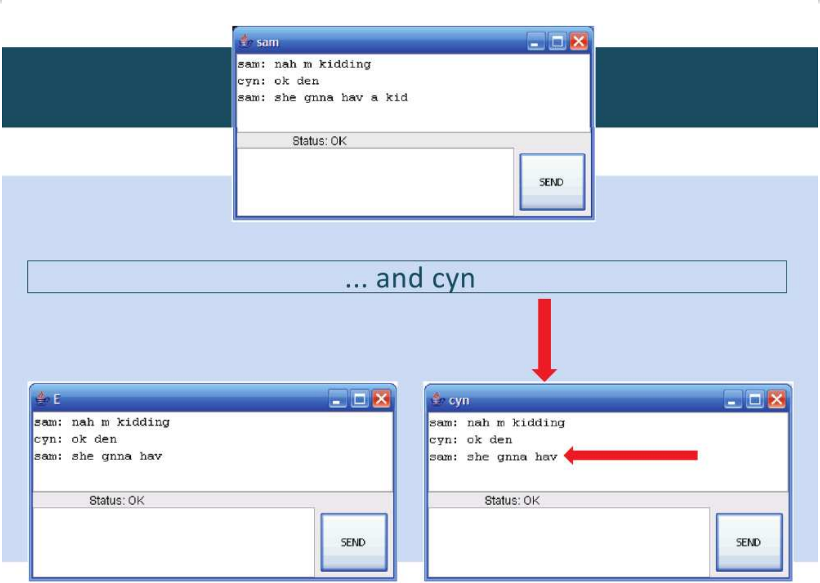


```
sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid
```

First part of SU relayed to E ...

```
sam: nah m kidding
cyn: ok den
sam: she gnna hav
```

```
sam: nah m kidding
cyn: ok den
```

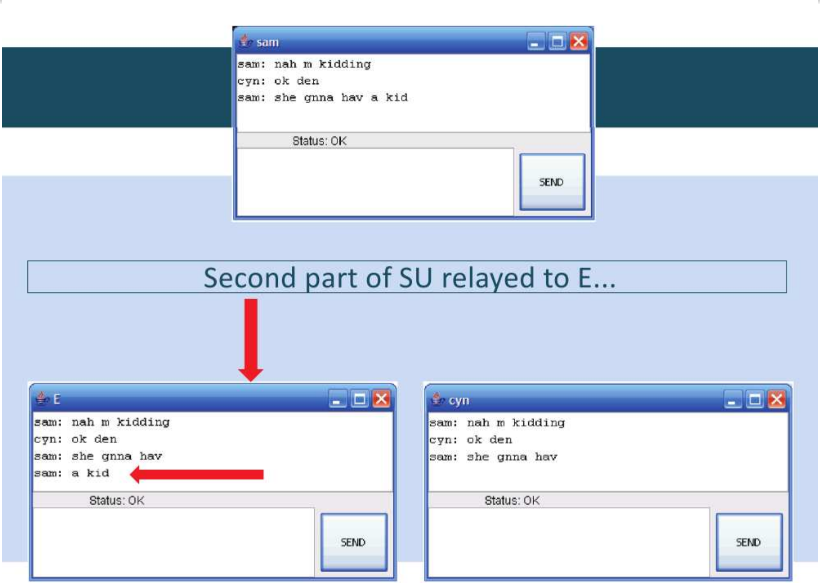


sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid

... and cyn

sam: nah m kidding
cyn: ok den
sam: she gnna hav

sam: nah m kidding
cyn: ok den
sam: she gnna hav

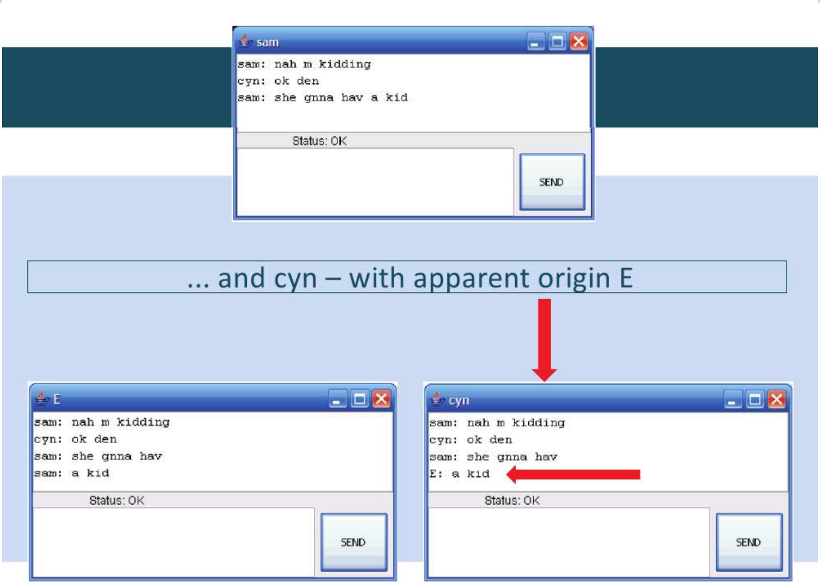


```
sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid
```

Second part of SU relayed to E...

```
sam: nah m kidding
cyn: ok den
sam: she gnna hav
sam: a kid
```

```
sam: nah m kidding
cyn: ok den
sam: she gnna hav
```



```
sam: nah m kidding
cyn: ok den
sam: she gnna hav a kid
```

... and cyn – with apparent origin E

```
sam: nah m kidding
cyn: ok den
sam: she gnna hav
sam: a kid
```

```
sam: nah m kidding
cyn: ok den
sam: she gnna hav
E: a kid
```

Experimental Study: An example

- ‘Bancil’ types:

  the only loss here is a pilot and a father which is kinda bad but someones gotta go

- ‘Aryan’ sees (AA):

  Bancil: the only loss here is a pilot and a father
  Bancil: which is kinda bad but someones gotta go

- ‘efparxng’ sees (AB):

  Bancil: the only loss here is a pilot and a father
  Aryan: which is kinda bad but someones gotta go

Experimental Study: Results

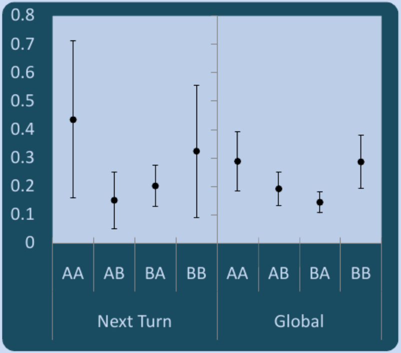
- We can observe: typing time of turn, number of ‘deletes’ used
  - *next turn* effects: the next participant to type
  - *global* effects: all participants turns until next intervention
- (We *can’t* observe time to start typing)

Experimental Study: Results

- We can observe: typing time of turn, number of 'deletes' used
    - *next turn* effects: the next participant to type
    - *global* effects: all participants turns until next intervention
- (We *can't* observe time to start typing)
- We can compare: speaker switch (AA/BB vs. AB/BA)
- We can compare: floor change (AA/BA vs. BB/AB)
- We can compare: first/second part coherence (Y/N)

Experimental Study: Results

- Main effect of *speaker switch* on number of 'deletes'

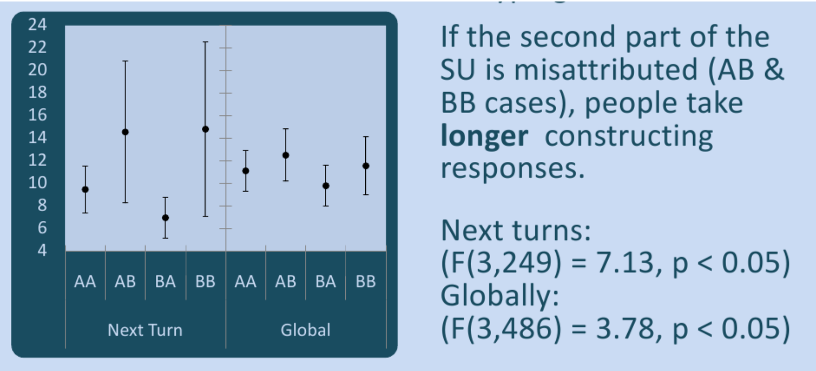


If the SU appears to be a cross-person one (AB & BA cases), people use **fewer** deletes in their responses.

Next turns:
(F(3,249) = 6.26, $p < 0.05$)
Globally:
(F(3,486) = 9.23, $p < 0.05$)

Experimental Study: Results

- Main effect of *floor change* on typing time of turn

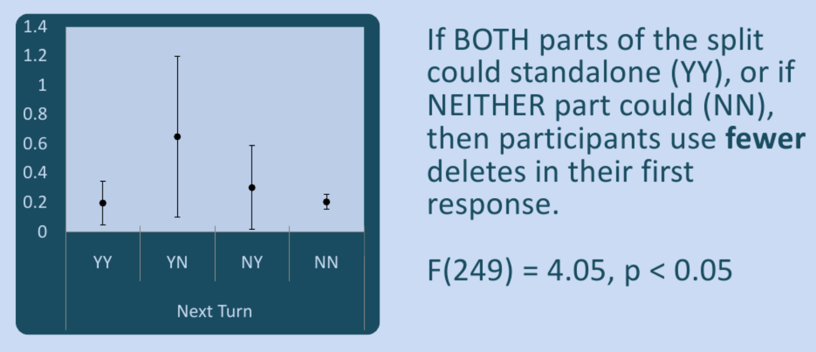


If the second part of the SU is misattributed (AB & BB cases), people take **longer** constructing responses.

Next turns:
(F(3,249) = 7.13, p < 0.05)
Globally:
(F(3,486) = 3.78, p < 0.05)

Experimental Study: Results

- Interaction effect of *1st- x 2nd-part coherence* on 'deletes'



If BOTH parts of the split could standalone (YY), or if NEITHER part could (NN), then participants use **fewer** deletes in their first response.

$F(249) = 4.05, p < 0.05$

Experimental Study: Conclusions

- Lack of speaker-switch effect on typing time suggests ease of processing
- Effect on deletes may be due to apparent party formation?
- Effect of floor change may be due to interference in turn-taking organisation
- Effect of 1st/2nd-part coherence suggests "garden-path"-style revision
- We're worried about the robustness of the setup . . .
    - . . . and we'd really like to know about onset delay . . .
    - . . . a character-by-character version is almost complete

Outline

1. Dialogue and Incrementality
   - Split Utterances and Alignment
2. Dynamic Syntax (DS)
   - A Quick Introduction to DS
   - DS and Dialogue Modelling
3. Empirical Investigations
   - Priming - Corpus Study
   - Split Utterances - Corpus Study
   - Split Utterances - Experiments
4. Dynamic Syntax & Type Theory with Records (TTR)
   - Adding TTR to DS
   - Fragments & Split Utterances in DS/TTR

DS and TTR: Motivation

- So far, we're happy that we're going in roughly the right direction:
  - Split utterances seem to fit the DS approach (mostly)
  - Priming results fit with prediction (so far as we can tell)

DS and TTR: Motivation

- So far, we're happy that we're going in roughly the right direction:
    - Split utterances seem to fit the DS approach (mostly)
    - Priming results fit with prediction (so far as we can tell)
- For a proper treatment of NSUs and SUs, DS needs:
    - Utterance function (speech acts?)
    - Responsibility for a (sub-)utterance (speaker, hearer?)
    - So we need more structured representations
- Want to avoid *forcing* this into all representations . . .
    - What should really be in the grammar?

Type Theory With Records

- See [Betarte and Tasistro, 1998], following Martin-Löf
- *Records* are sequences of label/value pairs:

$$\begin{bmatrix} l_1 & = & v_1 \\ l_2 & = & v_2 \\ l_3 & = & v_3 \end{bmatrix}$$

- *Record types* are sequences of label/type pairs:

$$\begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2 \\ l_3 & : & T_3 \end{bmatrix}$$

- Record types are true iff they are *inhabited/witnessed*
    - = there exists at least one record of that type
    - = successful type judgements for each label/value pair:

$$v_1 : T_1, \quad v_2 : T_2, \quad v_3 : T_3$$

Type Theory With Records

- Types can be *dependent* on earlier (higher-up) types:

$$\begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2(l_1) \\ l_3 & : & T_3(l_1, l_2) \end{bmatrix}$$

- We can have *nested* records and record types:

$$\begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & \begin{bmatrix} l'_1 & : & T'_1 \\ l'_2 & : & T'_2 \end{bmatrix} \\ l_3 & : & T_3(l_1, l_2.l'_1, l_2.l'_2) \end{bmatrix}$$

- We can have *functional* record types:

$$\lambda r : \begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2 \end{bmatrix} (\begin{bmatrix} l_3 & : & T_3 \\ l_4 & : & T_4(r.l_1, r.l_2) \end{bmatrix})$$

Type Theory With Records

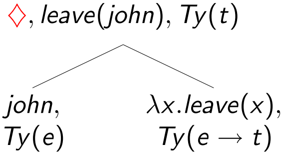
- Used for sentential semantics, e.g. [Cooper, 2005]
  - “A man left”: $\begin{bmatrix} x & : & \mathit{man} \\ p & : & \mathit{leave}(x) \end{bmatrix}$
  - for truth: x must be a man, p a proof that x left
  - “Every man left”: $\lambda r : \begin{bmatrix} x & : & \mathit{man} \end{bmatrix} (\begin{bmatrix} p & : & \mathit{leave}(r.x) \end{bmatrix})$
- Similarities to DRT representation: $\begin{array}{|c|}\hline x \\ \hline \mathrm{man}(x) \\ \mathrm{leave}(x) \\ \hline \end{array}$
- Used for dialogue modelling in the information-state-based tradition
  - [Cooper and Ginzburg, 2002, Ranta and Cooper, 2004, Fernández, 2006, Ginzburg, ming]

The best of both worlds?

- TTR gives us a type-theoretic framework, applicable to dialogue phenomena
- DS gives us an incremental framework using type theory as an underlying mechanism
- Can we combine the two?

The best of both worlds?

- TTR gives us a type-theoretic framework, applicable to dialogue phenomena
- DS gives us an incremental framework using type theory as an underlying mechanism
- Can we combine the two?



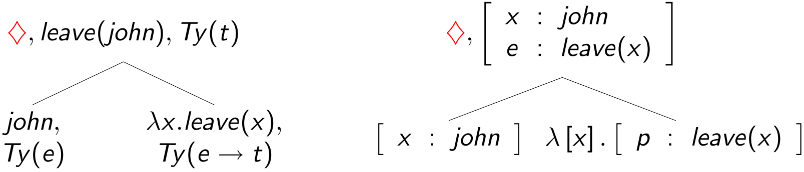
$\diamondsuit$, $leave(john)$, $Ty(t)$

$john$, $Ty(e)$

$\lambda x.leave(x)$, $Ty(e \rightarrow t)$

The best of both worlds?

- TTR gives us a type-theoretic framework, applicable to dialogue phenomena
- DS gives us an incremental framework using type theory as an underlying mechanism
- Can we combine the two?



$\Diamond, \mathit{leave}(\mathit{john}), \mathit{Ty}(t)$

$\mathit{john}, \mathit{Ty}(e)$

$\lambda x.\mathit{leave}(x), \mathit{Ty}(e \rightarrow t)$

$\Diamond, \left[\begin{array}{lcl} x & : & \mathit{john} \\ e & : & \mathit{leave}(x) \end{array}\right]$

$\left[\begin{array}{lcl} x & : & \mathit{john} \end{array}\right]$

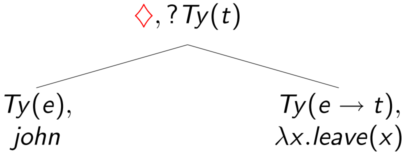
$\lambda\,[x]\,.\left[\begin{array}{lcl} p & : & \mathit{leave}(x) \end{array}\right]$

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types



$\Diamond, ?Ty(t)$

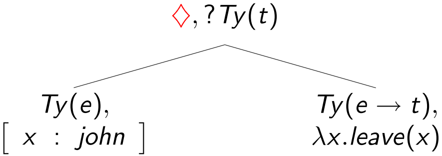
$Ty(e),$
$john$

$Ty(e \rightarrow t),$
$\lambda x.leave(x)$

IF	$?Ty(e)$
THEN	$\mathtt{put}(Ty(e))$ $\mathtt{put}(Fo(john))$
ELSE	$\mathtt{abort}$

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types



$\Diamond, ?Ty(t)$

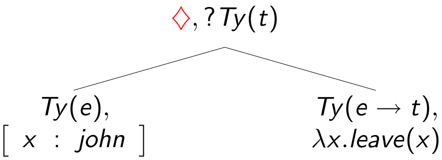
$Ty(e),$
$\left[\ x \ : \ john\ \right]$

$Ty(e \rightarrow t),$
$\lambda x.leave(x)$

IF $?Ty(e)$
THEN $\mathtt{put}(Ty(e))$
$\mathtt{put}(\left[\ x \ : \ john\ \right])$
ELSE $\mathtt{abort}$

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types
- Interpret *Ty*() simple type labels as referring to *final* TTR field type



$\Diamond, ?Ty(t)$

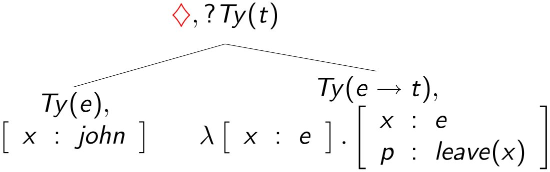
$Ty(e)$, $\left[\ x \ : \ john \ \right]$

$Ty(e \rightarrow t)$, $\lambda x.leave(x)$

```
IF      $?Ty(e)$
THEN    put($Ty(e)$)
        put($\left[\ x \ : \ john \ \right]$)
ELSE    abort
```

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types
- Interpret *Ty*() simple type labels as referring to *final* TTR field type



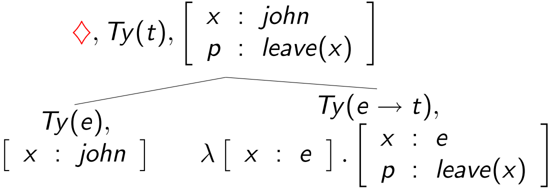
$$\Diamond, ?Ty(t)$$

$$Ty(e), \quad \left[\ x \ : \ john \ \right]$$

$$Ty(e \rightarrow t), \quad \lambda \left[\ x \ : \ e \ \right] . \left[\begin{array}{lcl} x & : & e \\ p & : & leave(x) \end{array} \right]$$

A simple version

- Replace *Fo*() epsilon-calculus labels with TTR record types
- Interpret *Ty*() simple type labels as referring to *final* TTR field type
- Function application as before for DS `elimination` process



$$\diamondsuit, Ty(t), \begin{bmatrix} x & : & john \\ p & : & leave(x) \end{bmatrix}$$

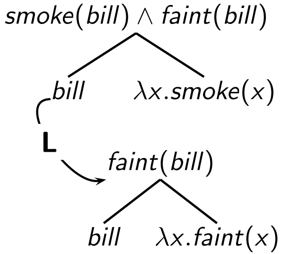
$$Ty(e), \begin{bmatrix} x & : & john \end{bmatrix}$$

$$Ty(e \rightarrow t), \lambda \begin{bmatrix} x & : & e \end{bmatrix} . \begin{bmatrix} x & : & e \\ p & : & leave(x) \end{bmatrix}$$

Adding in LINK relations

- For LINKed trees, we need conjunction

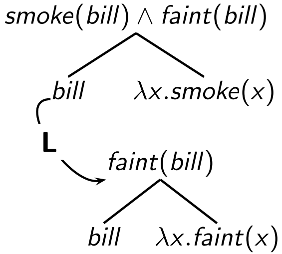
“Bill, who fainted, smokes.”



Adding in LINK relations

- For LINKed trees, we need conjunction
- Use *extension*: $\oplus$ where $r_1 \oplus r_2$ adds r_2 to the end of r_1
    - (for distinct labels; identical fields collapse [Cooper, 1998])

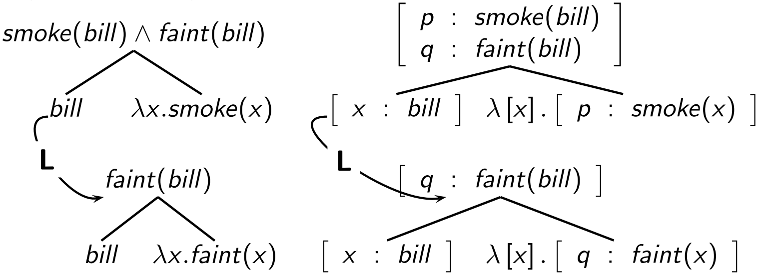
"Bill, who fainted, smokes."



Adding in LINK relations

- For LINKed trees, we need conjunction
- Use *extension*: $\oplus$ where $r_1 \oplus r_2$ adds r_2 to the end of r_1
  - (for distinct labels; identical fields collapse [Cooper, 1998])

“Bill, who fainted, smokes.”



$smoke(bill) \wedge faint(bill)$

$bill$ $\lambda x.smoke(x)$

L

$faint(bill)$

$bill$ $\lambda x.faint(x)$

$\begin{bmatrix} p & : & smoke(bill) \\ q & : & faint(bill) \end{bmatrix}$

$\begin{bmatrix} x & : & bill \end{bmatrix}$ $\lambda[x].\begin{bmatrix} p & : & smoke(x) \end{bmatrix}$

L

$\begin{bmatrix} q & : & faint(bill) \end{bmatrix}$

$\begin{bmatrix} x & : & bill \end{bmatrix}$ $\lambda[x].\begin{bmatrix} q & : & faint(x) \end{bmatrix}$

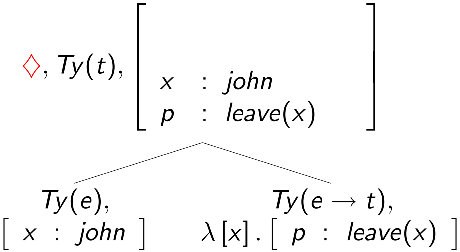
Can we do better?

- From an implementational point of view, this is OK
- But we're in danger of losing something
    - DS trees as they stand have a direct correspondence with semantics
    - Nodes are terms in the lambda-calculus
    - (Unreduced terms at daughter nodes)
    - What exactly are they now?
- Would prefer tree definitions via TTR(-compatible) logic
    - Type dependencies rather than abstraction (via [Kopylov, 2003] dependent intersection)
    - Initial versions for basic framework; LINK more complicated
    - (Meyer-Viol/White, forthcoming)

LINK as optional enrichment process

- Add utterance-event information
- Add speaker (or rather "responsible party") information

"John left"



$$\diamondsuit, Ty(t), \left[\begin{array}{lll} & & \\ x & : & john \\ p & : & leave(x) \end{array}\right]$$

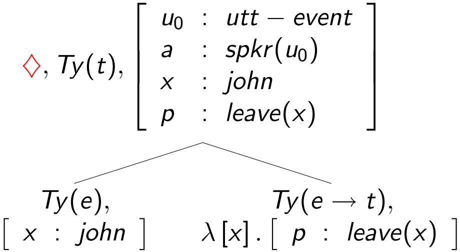
$$Ty(e), \quad \left[\begin{array}{lll} x & : & john \end{array}\right]$$

$$Ty(e \rightarrow t), \quad \lambda[x].\left[\begin{array}{lll} p & : & leave(x) \end{array}\right]$$

LINK as optional enrichment process

- Add utterance-event information
- Add speaker (or rather "responsible party") information

"John left"



$$\diamondsuit, Ty(t), \begin{bmatrix} u_0 & : & utt - event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix}$$

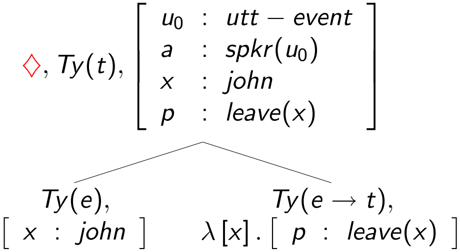
$$Ty(e), \quad \begin{bmatrix} x & : & john \end{bmatrix}$$

$$Ty(e \rightarrow t), \quad \lambda[x].\begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

LINK as optional enrichment process

- Add utterance-event information
- Add speaker (or rather "responsible party") information

"John left"



$$\diamondsuit, Ty(t), \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix}$$

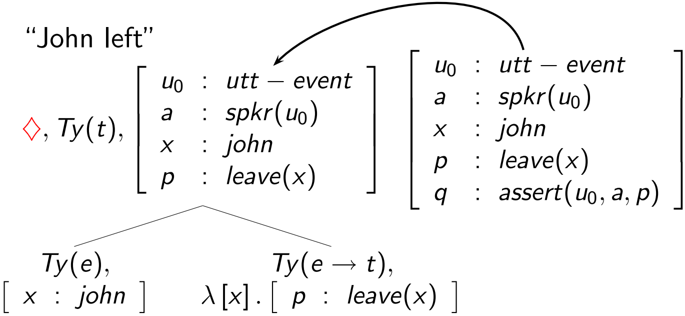
$$Ty(e), \quad \begin{bmatrix} x & : & john \end{bmatrix}$$

$$Ty(e \rightarrow t), \quad \lambda[x].\begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

- Allow *optional* inferences about speech acts

LINK as optional enrichment process

- Add utterance-event information
- Add speaker (or rather “responsible party”) information



“John left”

$$\diamondsuit, Ty(t), \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix} \quad \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q & : & assert(u_0, a, p) \end{bmatrix}$$

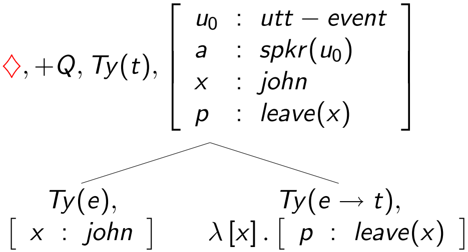
$$Ty(e), \begin{bmatrix} x & : & john \end{bmatrix} \qquad Ty(e \rightarrow t), \lambda[x].\begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

- Allow *optional* inferences about speech acts

LINK as optional enrichment process

- Speech act inferences conditional on syntax/semantics

"Did John leave?"



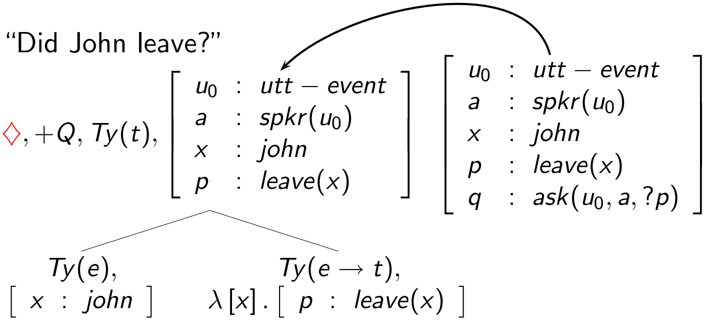
$$\diamondsuit, +Q, Ty(t), \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix}$$

$$Ty(e), \quad \begin{bmatrix} x & : & john \end{bmatrix}$$

$$Ty(e \rightarrow t), \quad \lambda [x] . \begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

LINK as optional enrichment process

- Speech act inferences conditional on syntax/semantics



“Did John leave?”

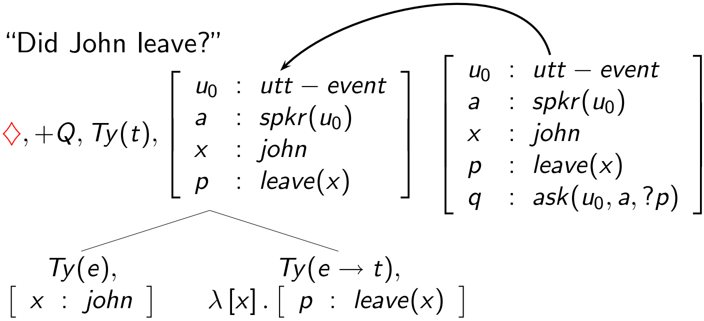
$$\diamondsuit, +Q, Ty(t), \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix} \quad \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q & : & ask(u_0, a, ?p) \end{bmatrix}$$

$$Ty(e), \quad \begin{bmatrix} x & : & john \end{bmatrix}$$

$$Ty(e \to t), \quad \lambda[x].\begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

LINK as optional enrichment process

- Speech act inferences conditional on syntax/semantics



"Did John leave?"

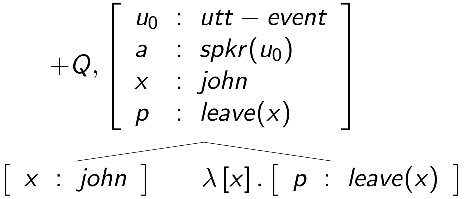
$$\diamondsuit, +Q, Ty(t), \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix} \quad \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q & : & ask(u_0, a, ?p) \end{bmatrix}$$

$$Ty(e), \begin{bmatrix} x & : & john \end{bmatrix} \qquad Ty(e \rightarrow t), \lambda[x].\begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

- Similarities with [Ginzburg et al., 2003]

An example: a "clausal" clarification request

A: "Did John leave?"

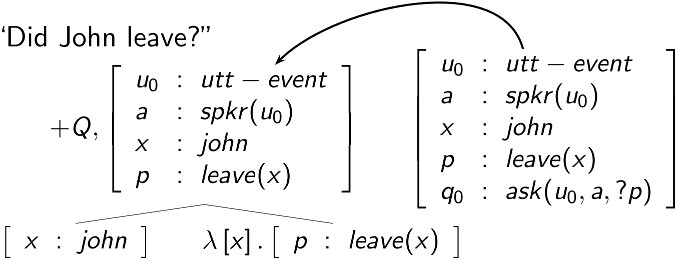


$$+Q, \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{array}\right]$$

$$\left[\ x\ :\ john\ \right] \qquad \lambda[x].\left[\ p\ :\ leave(x)\ \right]$$

An example: a “clausal” clarification request

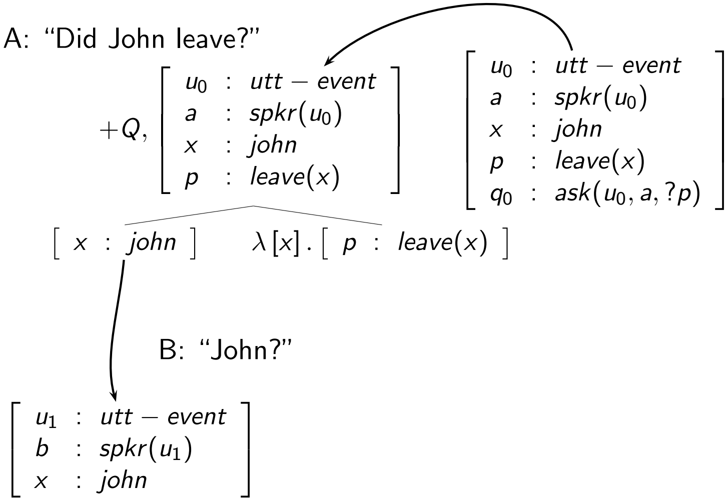
A: “Did John leave?”



$$+Q, \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{array}\right] \qquad \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q_0 & : & ask(u_0, a, ?p) \end{array}\right]$$

$$\left[\begin{array}{lcl} x & : & john \end{array}\right] \qquad \lambda[x].\left[\begin{array}{lcl} p & : & leave(x) \end{array}\right]$$

An example: a “clausal” clarification request



A: “Did John leave?”

$$+Q, \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{array}\right]$$

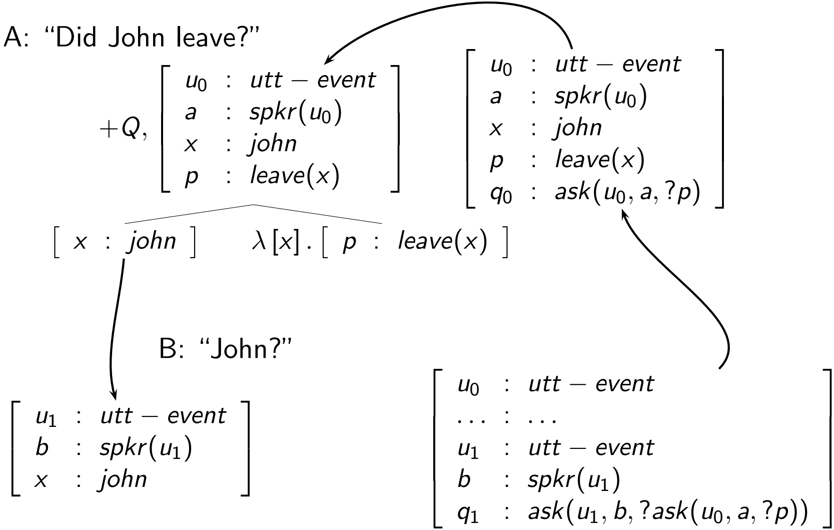
$$\left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q_0 & : & ask(u_0, a, ?p) \end{array}\right]$$

$$\left[\begin{array}{lcl} x & : & john \end{array}\right] \qquad \lambda[x].\left[\begin{array}{lcl} p & : & leave(x) \end{array}\right]$$

B: “John?”

$$\left[\begin{array}{lcl} u_1 & : & utt-event \\ b & : & spkr(u_1) \\ x & : & john \end{array}\right]$$

An example: a "clausal" clarification request



A: "Did John leave?"

$$+Q, \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{bmatrix}$$

$$\begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q_0 & : & ask(u_0, a, ?p) \end{bmatrix}$$

$$\begin{bmatrix} x & : & john \end{bmatrix} \qquad \lambda[x] . \begin{bmatrix} p & : & leave(x) \end{bmatrix}$$

B: "John?"

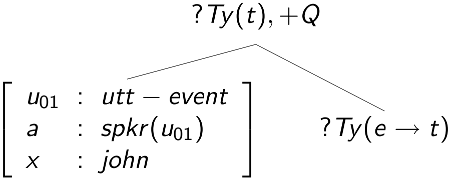
$$\begin{bmatrix} u_1 & : & utt-event \\ b & : & spkr(u_1) \\ x & : & john \end{bmatrix}$$

$$\begin{bmatrix} u_0 & : & utt-event \\ \ldots & : & \ldots \\ u_1 & : & utt-event \\ b & : & spkr(u_1) \\ q_1 & : & ask(u_1, b, ?ask(u_0, a, ?p)) \end{bmatrix}$$

An example: a “constituent” clarification request

- Add [Poesio and Traum, 1997]'s *micro-conversational events*

A: “Did John . . .”



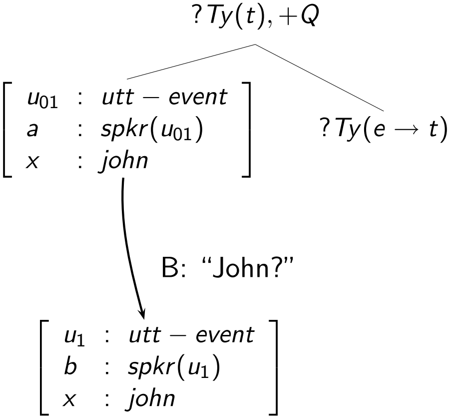
$?Ty(t), +Q$

$$\left[\begin{array}{lcl} u_{01} & : & utt-event \\ a & : & spkr(u_{01}) \\ x & : & john \end{array}\right] \qquad ?Ty(e \rightarrow t)$$

An example: a "constituent" clarification request

- Add [Poesio and Traum, 1997]'s *micro-conversational events*

A: "Did John . . ."



$?Ty(t), +Q$

$$\left[\begin{array}{lcl} u_{01} & : & utt-event \\ a & : & spkr(u_{01}) \\ x & : & john \end{array}\right]$$

$?Ty(e \rightarrow t)$

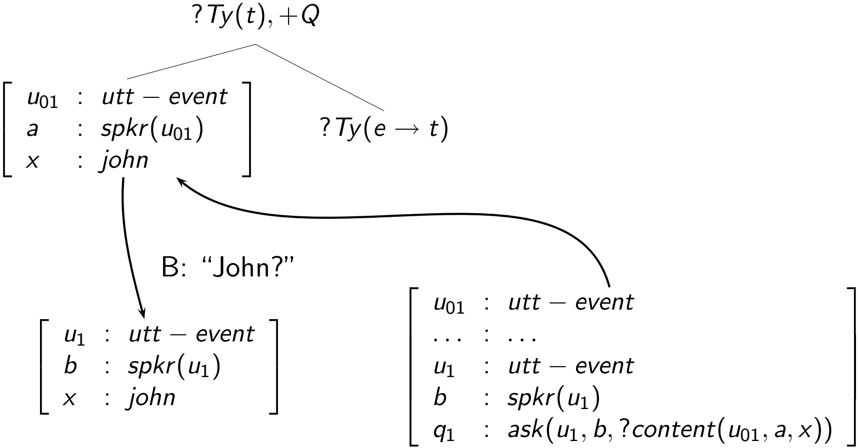
B: "John?"

$$\left[\begin{array}{lcl} u_1 & : & utt-event \\ b & : & spkr(u_1) \\ x & : & john \end{array}\right]$$

An example: a "constituent" clarification request

- Add [Poesio and Traum, 1997]'s *micro-conversational events*

  A: "Did John . . ."



$$?Ty(t), +Q$$

$$\begin{bmatrix} u_{01} & : & utt-event \\ a & : & spkr(u_{01}) \\ x & : & john \end{bmatrix} \qquad ?Ty(e \rightarrow t)$$

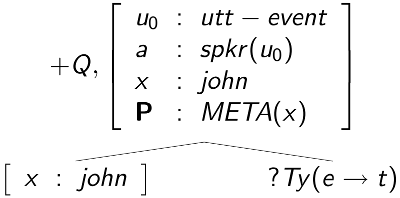
B: "John?"

$$\begin{bmatrix} u_1 & : & utt-event \\ b & : & spkr(u_1) \\ x & : & john \end{bmatrix}$$

$$\begin{bmatrix} u_{01} & : & utt-event \\ \ldots & : & \ldots \\ u_1 & : & utt-event \\ b & : & spkr(u_1) \\ q_1 & : & ask(u_1, b, ?content(u_{01}, a, x)) \end{bmatrix}$$

An example: a clarificational split utterance

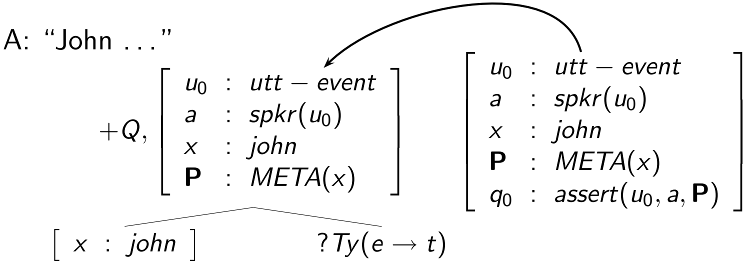
A: "John . . ."



$$+Q, \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ \mathbf{P} & : & META(x) \end{bmatrix}$$

$$\begin{bmatrix} x & : & john \end{bmatrix} \qquad ?Ty(e \rightarrow t)$$

An example: a clarificational split utterance



A: "John ..."

$$+Q, \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ \mathbf{P} & : & META(x) \end{bmatrix}$$

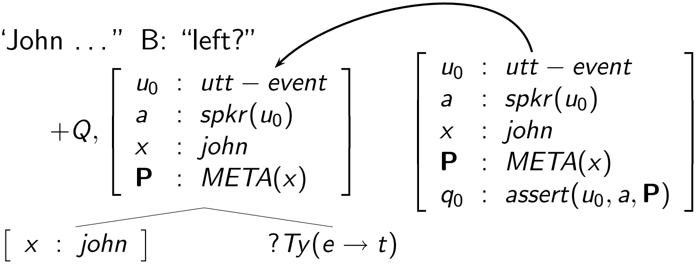
$$\begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ \mathbf{P} & : & META(x) \\ q_0 & : & assert(u_0, a, \mathbf{P}) \end{bmatrix}$$

$$\begin{bmatrix} x & : & john \end{bmatrix}$$

$$?Ty(e \rightarrow t)$$

An example: a clarificational split utterance

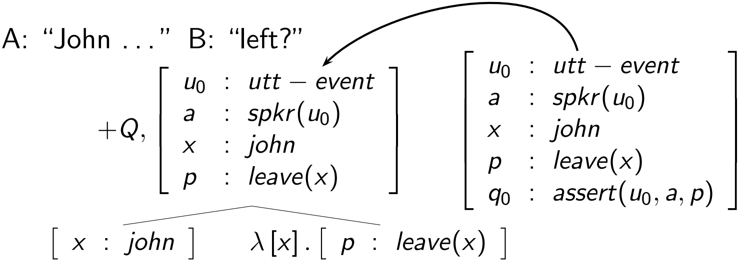
A: “John . . . ” B: “left?”



$$+Q, \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ \mathbf{P} & : & META(x) \end{bmatrix} \qquad \begin{bmatrix} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ \mathbf{P} & : & META(x) \\ q_0 & : & assert(u_0, a, \mathbf{P}) \end{bmatrix}$$

$$\begin{bmatrix} x & : & john \end{bmatrix} \qquad ?Ty(e \rightarrow t)$$

An example: a clarificational split utterance



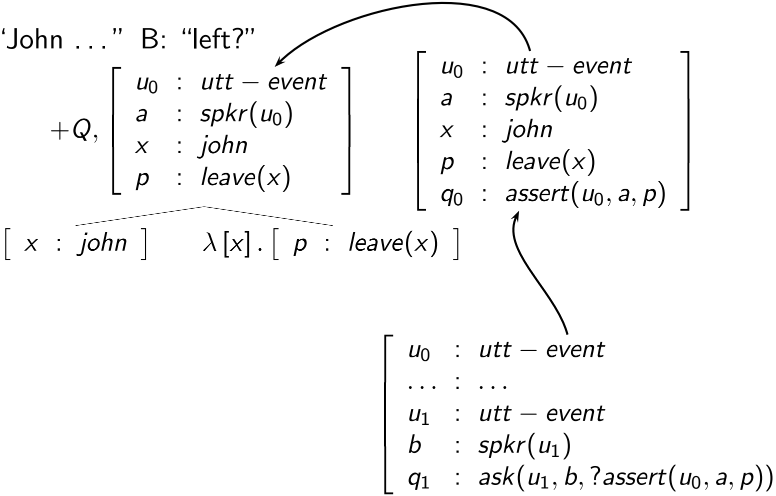
A: “John . . . ” B: “left?”

$$+Q, \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{array}\right] \qquad \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q_0 & : & assert(u_0, a, p) \end{array}\right]$$

$$\left[\begin{array}{lcl} x & : & john \end{array}\right] \qquad \lambda[x].\left[\begin{array}{lcl} p & : & leave(x) \end{array}\right]$$

An example: a clarificational split utterance

A: “John . . .” B: “left?”



$$+Q, \left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \end{array}\right]$$

$$\left[\begin{array}{lcl} x & : & john \end{array}\right] \qquad \lambda[x].\left[\begin{array}{lcl} p & : & leave(x) \end{array}\right]$$

$$\left[\begin{array}{lcl} u_0 & : & utt-event \\ a & : & spkr(u_0) \\ x & : & john \\ p & : & leave(x) \\ q_0 & : & assert(u_0, a, p) \end{array}\right]$$

$$\left[\begin{array}{lcl} u_0 & : & utt-event \\ \ldots & : & \ldots \\ u_1 & : & utt-event \\ b & : & spkr(u_1) \\ q_1 & : & ask(u_1, b, ?assert(u_0, a, p)) \end{array}\right]$$

(Eventual) Conclusions

- Incrementality of DS with the flexibility of TTR
- Core grammar essentially as before
- Optional enrichment processes for speech act information
    - similarities to [Ginzburg and Cooper, 2004] et al.
    - similarities to [Asher and Lascarides, 2003] et al.
- A proper treatment of split utterances . . . ?
    - capturing insights of [Poesio and Rieser, 2010]
    - more fundamentally incremental

Thanks!

And thanks to:

Pat Healey, Christine Howes, Graham White, Arash Eshghi, Greg Mills at QMUL

Ruth Kempson, Eleni Gregoromichelaki, Wilfried Meyer-Viol at KCL

Andrew Gargett in Saarbrücken, Ronnie Cann in Edinburgh, Yo Sato in Herts

Asher, N. and Lascarides, A. (2003).
Logics of Conversation.
Cambridge University Press.

Betarte, G. and Tasistro, A. (1998).
Extension of martin löf type theory with record types and subtyping.
In Sambin, G. and Smith, J., editors, *25 Years of Constructive Type Theory*. Oxford University Press.

Blackburn, P. and Meyer-Viol, W. (1994).
Linguistics, logic and finite trees.
Logic Journal of the Interest Group of Pure and Applied Logics, 2(1):3–29.

Branigan, H., Pickering, M., and Cleland, A. (2000).
Syntactic co-ordination in dialogue.
Cognition, 75:13–25.

Cooper, R. (1998).
Information states, attitudes and dependent record types.
In *Proceedings of ITALLC-98*, pages 85–106.

Cooper, R. (2005).
Records and record types in semantic theory.
Journal of Logic and Computation, 15(2):99–112.

Cooper, R. and Ginzburg, J. (2002).
Using dependent record types in clarification ellipsis.
In Bos, J., Foster, M., and Matheson, C., editors, *Proceedings of the 6th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 45–52, Edinburgh, Scotland.

Fernández, R. (2006).
Non-Sentential Utterances in Dialogue: Classification, Resolution and Use.
PhD thesis, King's College London, University of London.

Garrod, S. and Anderson, A. (1987).
Saying what you mean in dialogue: A study in conceptual and semantic co-ordination.
Cognition, 27:181–218.

Ginzburg, J. (forthcoming).
The Interactive Stance: Meaning for Conversation.
Studies in Computational Linguistics. CSLI Publications.
Draft chapters available from:
`http://www.dcs.kcl.ac.uk/staff/ginzburg`.

Ginzburg, J. and Cooper, R. (2004).
Clarification, ellipsis, and the nature of contextual updates in dialogue.
Linguistics and Philosophy, 27(3):297–365.

Ginzburg, J., Sag, I., and Purver, M. (2003).

Integrating conversational move types in the grammar of conversation.
In Kühnlein, P., Rieser, H., and Zeevat, H., editors, *Perspectives on Dialogue in the New Millennium*, volume 114 of *Pragmatics and Beyond New Series*, pages 25–42. John Benjamins.

Hardt, D. (2008).
VP Ellipsis and Constraints on Interpretation.
Cambridge University Press, Cambridge.

Healey, P., Purver, M., King, J., Ginzburg, J., and Mills, G. (2003).
Experimenting with clarification in dialogue.
In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*, Boston, Massachusetts.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001).

Dynamic Syntax: The Flow of Language Understanding.
Blackwell.

Lerner, G. (2004).
Collaborative turn sequences.
In *Conversation analysis: Studies from the first generation*, pages 225–256. John Benjamins.

Moschitti, A. (2006).
Making tree kernels practical for natural language learning.
In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Pickering, M. and Ferreira, V. (2008).
Structural priming: A critical review.
Psychological Bulletin, 134(3):427–459.

Poesio, M. and Rieser, H. (2010).
Completions, coordination, and alignment in dialogue.

Dialogue and Discourse, 1:1–89.

- Poesio, M. and Traum, D. (1997).
  Conversational actions and discourse situations.
  Computational Intelligence, 13(3).

- Ranta, A. and Cooper, R. (2004).
  Dialogue systems as proof editors.
  Journal of Logic, Language and Information, 13:145–189.

- Reitter, D., Moore, J., and Keller, F. (2006).
  Priming of syntactic rules in task-oriented dialogue and spontaneous conversation.
  In Sun, R., editor, *Proceedings of the 28th Conference of the Cognitive Science Society*, pages 685–690.

- Rühlemann, C. (2007).
  Conversation in context: a corpus-driven approach.
  Continuum.

Schlangen, D. and Skantze, G. (2009).
A general, abstract model of incremental dialogue processing.
In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 710–718, Athens, Greece. Association for Computational Linguistics.

Schuler, W., Wu, S., and Schwartz, L. (2009).
A framework for fast incremental interpretation during speech decoding.
Computational Linguistics, 35(3):313–343.

Skuplik, K. (1999).
Satzkooperationen. definition und empirische untersuchung.
SFB 360 1999/03, Bielefeld University.

Szczepek, B. (2000).
Formal Aspects of Collaborative Productions in English Conversation.

Interaction and Linguistic Structures (InLiSt), *http://www.uni-potsdam.de/u/inlist/issues/17/*.