DS-TTR and Incremental Dialogue Processing

Julian Hough
with Matthew Purver, Arash Eshghi,
Ruth Kempson and Eleni Gregoromichelaki

Cognitive Science Group
School of Electronic Engineering and Computer Science
Queen Mary University of London

IWTTR-1, Gothenburg. 19th of December, 2012

Problem 1: Mid-utterance self-repairs

General form:

utterance [*reparandum* + {*interregnum*}*repair*]*continuation* [Shriberg, 1994, onwards]

"Flights to [London, + {uhh,} Paris] on Tuesday"
[replacement, constructed example]

"I [love, + {I mean,} really love] Haskell"
[insertion, constructed example]

" [[I guess + I c-,] + I think] it's got some relevance, "
[embedded, Switchboard sw4330]

"I saw John today [+ {uhh} in town]"
[extension, constructed example]

Problem 1: Mid-utterance self-repairs

"[the interview was $+\{\ldots\}$ it was] alright"

[Clark, 1996, p.266]

"Peter went [swimming with Susan + {or rather} surfing], on Tuesday." *[Semdial reviewer]*

- [Brennan and Schober, 2001] show people use the reparandum to help subjects make faster decisions:

  "Pick the yell-purple square" *faster*

  "Pick the uhh-purple square"

Problem 1: Mid-utterance self-repairs

- SRs can potentially occur anywhere in the utterance and of varying length.
  - [Shriberg and Stolcke, 1998]- steady exponential decay of likeliness of retrace getting one word longer with each word in utterance.
- *Many* surface forms; not just repetitions- insertion, substitution and complex, ‘hybrid’ types [Shriberg, 1994]. Lots of them! Every 30 words, or every 4.2 utterances in Switchboard.
- Dialogue participants/systems cannot/should not ‘delete’ the reparandum!
- People aware of nervousness/hesitation.

Problem 2: Compound contributions

Daughter: Oh here dad, a good way to get those corners out
Dad: is to stick yer finger inside.
Daughter: well, that's one way *[Lerner, 1991]*

B: Did you burn...
A: myself? No, fortunately not.

Problem 2: Compound Contributions

A: "so, uh, I'm terrified to speak [in, + { uh } – "
B: "in France]"
A: "in France]" *[Switchboard]*

- Hearer-speaker role reversal half way through syntactic construction, fluent or otherwise.
- Potential change in dialogue act and indexical pronoun resolution.
- Utterances extend syntactically complete utterances, very frequent in corpora [Purver et al, 2009]

Solution: Incremental dialogue processing

- **Parsing** must operate on-line (left-right, word-by-word at least) and semantically

  [Milward, 1994]'s properties of *strong incremental interpretation* (maximal semantic content) and incremental representation (each word's semantic representation recoverable)

- **Generation** should be able to operate with partial input

  Should be able to change goal input mid-utterance

- **Both modules** should be able to take over from one another (and incremental context)

  Both need to maintain downgraded representations (reparanda)

Solution: Incremental dialogue processing

- **Dialogue management/framework** should be able to interface with parser and generator on-line.

  Should add contextual speech act information to the representations.

  Suitable procedural context model needed (of the actions involved in parsing and generation.)

  Inference should be more efficient if the parser and generator are working on shared structures.

Previous work: Incremental Parsing

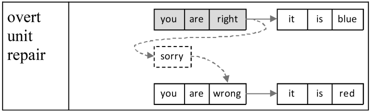
- Incremental interpretation using CCG [Milward, 1995] required extra machinery
- Left-right word-by-word parsing: Recent progress with PLTAG [Demberg and Keller, 2008]
  - Requires extra prediction and verification rules to yield connected structure
  - Semantics interface still not clearly defined
- Model of procedural context still in the parser, will become important later.
- Interchangeability with generation not well defined, though see [Neumann, 1998]

Previous work: Incremental Generation

- [Kempen and Hoenkamp, 1987][De Smedt 91]- incremental generation and self-repair. Insertion of structure allowed directly into existing trees.
  - "John was working in the lab - seemed to be working in the lab"
  - Allows special rules to operate on trees for insertion. Aim to make syntactic tree construction efficient and modifiable.
  - Integration with parsing? Semantics/origin of LF? Tracking reparandum?
- [Guhe, 2007]- optimizing representation for LF
  - Incremental and tree-based representation. Impasse in syntactic formulation caused by new LF causes self-repair.
  - Parsing integration/Interchangeability? Syntax blind to semantics.

Previous Work: Dialogue Systems- generating disfluency

- [Skantze and Hjalmarsson, 2010]'s *Jindigo* generates repairs by speech plan comparison (form of *self-monitoring*:



- Speech segments = one-to-one input concept-string correspondences- scalability/parsing integration difficult.
- [Buß and Schlangen, 2011]'s dialogue management strategy within IU framework. UNDO as a dialogue move.
  - Only for generation, triggered by revoked input.
  - Input IUs are words and output IUs frames- no procedural units.

Previous work: Dialogue Models

- **Self-repairs**- Ginzburg et al., 2007- mid-utterance repairs treated like resolution of clarification requests (CRs).
  - Edit signal equivalent to CR (although not necessarily explicit one).
  - PENDING component updated word-by-word- content can be self-queried, QUDs added and resolved.
  - Gives handle on the dialogue semantics of self-repair.
  - Procedural context still over units larger than the word (MOVES component in the dialogue gameboard).
  - Needs to be interfaced with incremental grammar/ parsing or generation mechanisms.
- **CCs**- [Poesio and Rieser 2010] detailed plan recognition- inflexible?

SUMMARY: what needs to be addressed...

- *Parsing*: lack of fully incremental processing account. Deletion/ignoring of reparandum in self-repairs. No discourse model.
- *Generation*: lack of full integration with dialogue manager (incremental access to representations)/discourse model neither/nor genuine inter-changeability with parsing- cross-person CC's.
- *Dialogue models/systems*: lack of integration with incremental grammars.
- Generally- no way of accounting for disfluency
- How much of the mechanisms for self-repairs and CCs can be left just to the parsing/generation modules?

What we need. . . Incremental stuff at all levels

Dynamic Syntax + TTR + IU Framework/Jindigo

- An incremental grammar formalism
  - *Dynamic Syntax* [Kempson et al., 2001]
- Interface between incremental representations and domain semantics
  - *Type Theory with Records (TTR)* [Cooper, 2005]
- An incremental dialogue framework which can store procedural context
  - *Incremental Unit (IU)* framework [Schlangen and Skantze, 2009]
  - *Jindigo* [Skantze and Hjalmarsson, 2010]

Challenge: Optimising representations and mechanisms.

Dynamic Syntax

- DS grammar encodes the word-by-word incremental growth of semantic representations directly.
- No independent layer of syntactic processing.
- Grammaticality is defined in terms of parsability in sequence.
- DS is bidirectional, i.e. generation is parasitic on parsing. *Self-monitoring* comes for free.
- *Parsing actions* (lexical and computational actions) are first class citizens of the grammar.
- Using DS context it is possible to model compound contributions, adjuncts, clarification requests, fragments and ellipsis- [Gargett et al., 2009, Gregoromichelaki et al., 2009, Purver et al., 2010] inter alia.

Similarities

DS trees	TTR records/record types
Tree subsumption relation	Subtype relation
Tree monotonic growth	Subtyping operations
Link adjunction	Meet type/merging
Unfixed nodes	Unbound variables
Formula values	Manifest fields

- Advantages of TTR for DS:
    - more fine-grained semantics than current FOL formulae
    - DS tree representations can interface with non-linguistic contextual information in the system
      e.g. Dialogue move/illocutionary force can be interpreted word-by-word by dialogue manager
- We still need DS's notion of procedural/processing context

Enriching DS lexical actions

- Context dependent values can be formally defined now [Purver et al., 2010]

myself :

$$
\begin{array}{ll}
\text{IF} & ?Ty(e), r : \left[\begin{array}{lcl} ctxt & : & \left[\begin{array}{lcl} u & : & utt \\ x & : & e \\ p_{=spkr(u,x)} & : & t \end{array} \right] \end{array} \right], \\
 & \uparrow_0\uparrow_{1*}\downarrow_0 r1 : \left[\begin{array}{lcl} cont & : & \left[\begin{array}{lcl} x1_{=r.ctxt.x} & : & e \end{array} \right] \end{array} \right] \\
\text{THEN} & \mathtt{put}(Ty(e)), \\
 & \mathtt{put}(r \wedge \left[\begin{array}{lcl} cont & : & \left[\begin{array}{lcl} x_{=r.ctxt.x} & : & e \end{array} \right] \end{array} \right]) \\
\text{ELSE} & \mathtt{abort}
\end{array}
$$

- Start to look more like TTR functions
- Use of dependent record types. Use of paths.

DS-TTR parsing and generation

- Recent variant uses TTR *record types* on the trees [Purver et al., 2011].
- Record type compilation for *partial trees* [Hough, 2011] allows strong incremental interpretation [Milward, 1994].
- Record types can be compared to domain concepts through *subtype/supertype* relation checking.
- In generation, a goal tree in former DS generation [Purver and Kempson, 2004] can be a TTR *goal concept* (record type); less tied to grammar.
- Faster *pre-verbal lexicalisation* for DS generation (forthcoming)

Incremental DS-TTR parsing

Parsing *Robin arrives*:

$$\begin{bmatrix} x & : & e \\ p & : & t \end{bmatrix}$$

Incremental DS-TTR parsing

Parsing *Robin arrives*:

Robin

$$\begin{bmatrix} x_{=robin} & : & e \\ p & : & t \end{bmatrix}$$

Incremental DS-TTR parsing

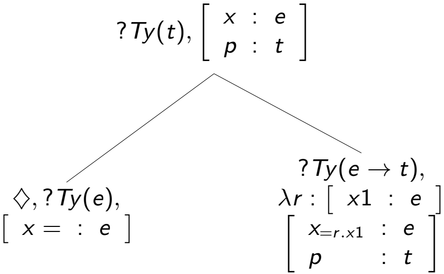
Parsing *Robin arrives*:

Robin arrives

$$\begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

Incremental DS-TTR parsing

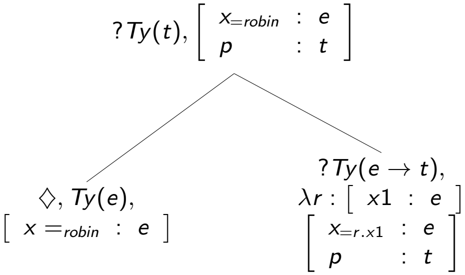
Parsing *Robin arrives*:



Incremental DS-TTR parsing

Parsing *Robin arrives*:

Robin



Incremental DS-TTR parsing

Parsing *Robin arrives*:

Robin

$$?Ty(t), \begin{bmatrix} x_{=robin} & : & e \\ p & : & t \end{bmatrix}$$

$$Ty(e), \begin{bmatrix} x =_{robin} & : & e \end{bmatrix}$$

$$\Diamond, ?Ty(e \rightarrow t), \lambda r : \begin{bmatrix} x1 & : & e \end{bmatrix} \begin{bmatrix} x_{=r.x1} & : & e \\ p & : & t \end{bmatrix}$$

Incremental DS-TTR parsing

Parsing *Robin arrives*:

Robin arrives

$$\Diamond, Ty(t), \left[\begin{array}{lcl} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{array} \right]$$

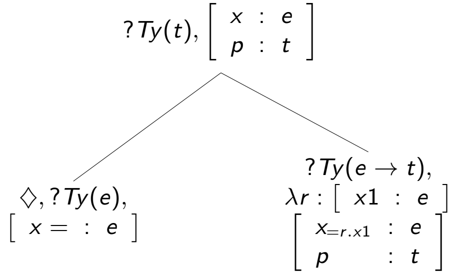
$$Ty(e), \left[\; x =_{robin} \; : \; e \; \right]$$

$$Ty(e \rightarrow t), \lambda r : \left[\; x1 \; : \; e \; \right] \left[\begin{array}{lcl} x_{=r.x1} & : & e \\ p_{=arrive(x)} & : & t \end{array} \right]$$

Incremental DS-TTR generation

Generating *Robin arrives*:

$$\text{GOAL}:\ \begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}\ \text{SUBTYPE}$$



Incremental DS-TTR generation

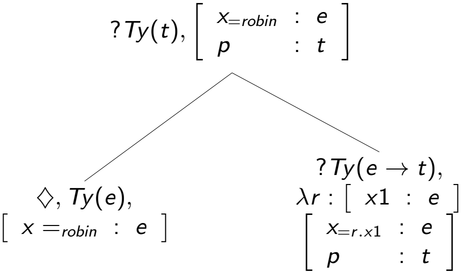
Generating *Robin arrives*:

Robin

GOAL :

$$\begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

SUBTYPE



Incremental DS-TTR generation

Generating *Robin arrives*:

Robin

GOAL :

$$\begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

SUBTYPE

$$?Ty(t), \begin{bmatrix} x_{=robin} & : & e \\ p & : & t \end{bmatrix}$$

$$Ty(e), \begin{bmatrix} x =_{robin} & : & e \end{bmatrix}$$

$$\Diamond, ?Ty(e \rightarrow t), \lambda r : \begin{bmatrix} x1 & : & e \end{bmatrix} \begin{bmatrix} x_{=r.x1} & : & e \\ p & : & t \end{bmatrix}$$

Incremental DS-TTR generation

Generating *Robin arrives*:

Robin arrives

GOAL :

$$\begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

MATCHES!

$$\Diamond, Ty(t), \begin{bmatrix} x_{=robin} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

$$Ty(e), \begin{bmatrix} x =_{robin} & : & e \end{bmatrix}$$

$$Ty(e \rightarrow t), \lambda r : \begin{bmatrix} x1 & : & e \end{bmatrix} \begin{bmatrix} x_{=r.x1} & : & e \\ p_{=arrive(x)} & : & t \end{bmatrix}$$

DS-TTR parsing context as a DAG

- Parsing starts from partial tree, reads in words one-by-one applying corresponding DS parsing actions.
- This process is modelled on a Directed Acyclic Graph (DAG) [Purver et al., 2011, Sato, 2011] where:
    - Nodes = Trees
    - Edges = DS parsing actions
    - Different paths represent different parsing strategies.
- Can model ambiguity [Sato, 2011]
- Can be integrated into the IU framework [Purver et al., 2011]

DyLan dialogue framework and system

- DyLan parser [Purver et al., 2011] and generator are modules in Jindigo [Skantze and Hjalmarsson, 2010], based on the IU framework [Schlangen and Skantze, 2009]
- Uses the graph-based input and output buffers.
- Uses a DS-TTR parsing DAG in both modules.
- The notions of *GroundedIn* links between IUs, *commitment*, and *revoking* IUs.

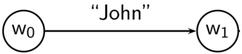
DyLan dialogue framework and system

- **Parsing module**:
  - *Input IUs:* Words from ASR
  - *Processing:* Constructs a DS-TTR parsing DAG, GroundedIn corresponding words
  - *Output IUs:* TTR record types (concepts) to dialogue manager, GroundedIn corresponding path of the DS-TTR DAG

DyLan parsing

John

WORD GRAPH
(INPUT)

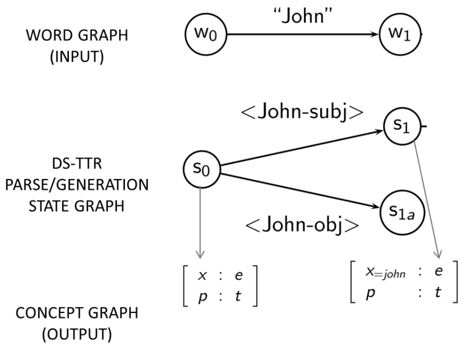


DS-TTR
PARSE/GENERATION
STATE GRAPH

CONCEPT GRAPH
(OUTPUT)

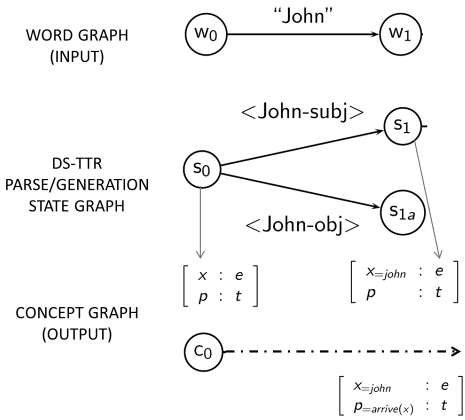
DyLan parsing

John

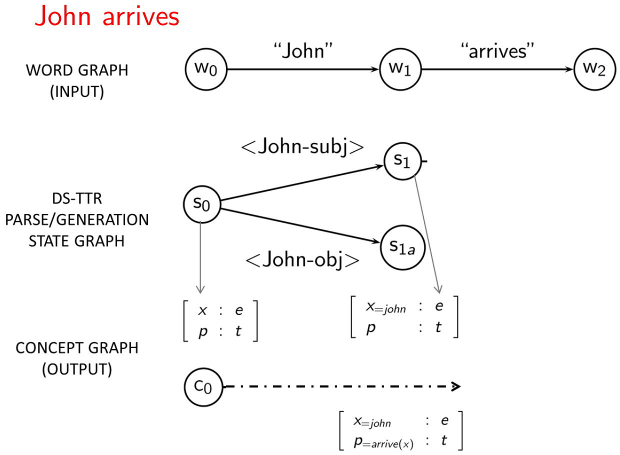


DyLan parsing

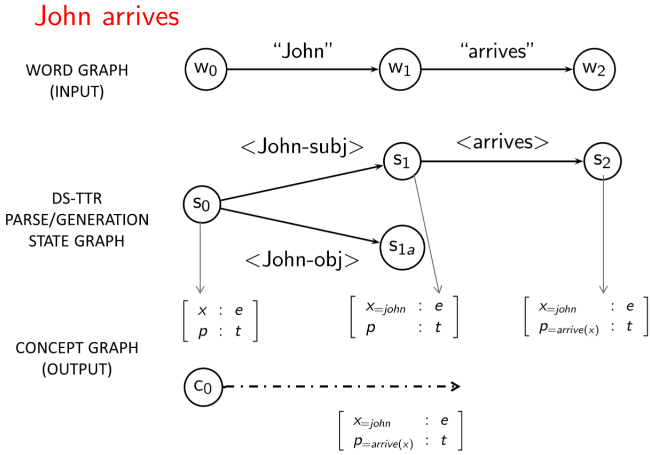
John



DyLan parsing

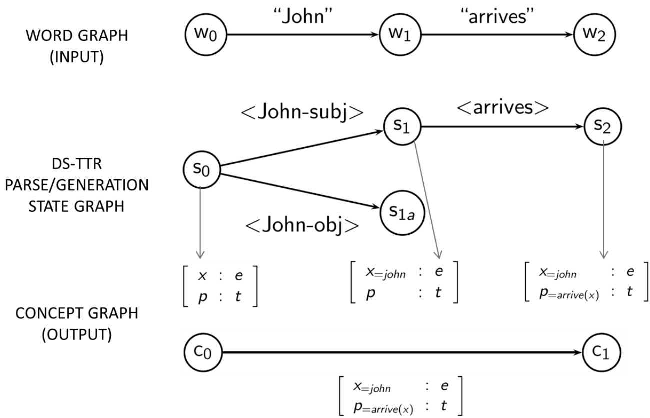


DyLan parsing



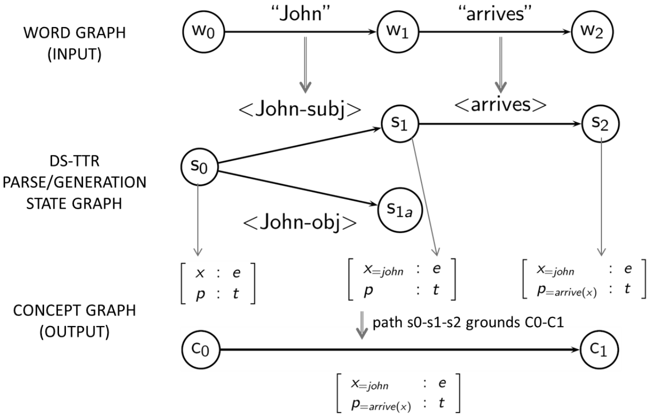
DyLan parsing

John arrives



DyLan parsing

John arrives



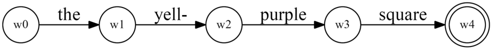
Parsing self-repairs incrementally

- Parses word-by-word incrementally, compiles TTR formulae.
- Subtype checks TTR formulae against domain concepts.
- Adds TTR formulae if there is a valid subtype in the domain concepts.
- If parse fails or no valid subtype in domain concepts: `REPAIR`:
  - Backtrack along DAG's word edges until successful parse and valid subsumption of a domain concept.

Parsing self-repairs incrementally

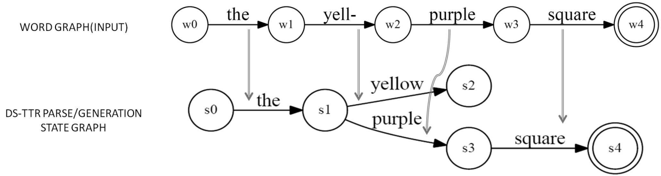
USER: "The yell-purple square"

WORD GRAPH(INPUT)



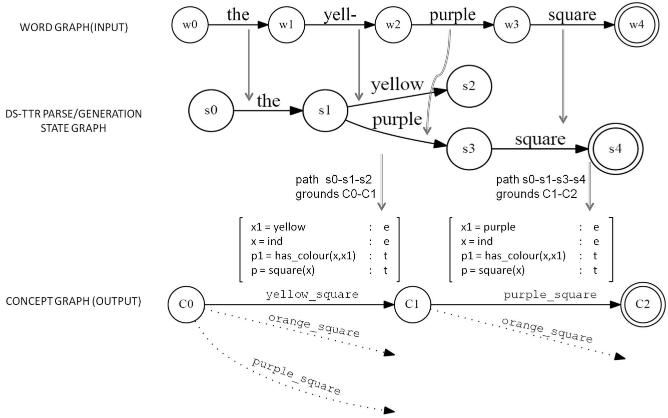
Parsing self-repairs incrementally

USER: "The yell-purple square"



Parsing self-repairs incrementally

USER: "The yell-purple square"



DyLan generation

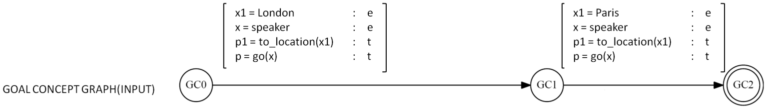
- **Generation module**:
- *Input IUs:* TTR record types goal concepts from dialogue manager.
- *Processing:* Constructs *the same* DS-TTR parsing DAG as parsing module, GroundedIn goal concept graph.
- *Output IUs:* Words (speech units), GroundedIn paths in the DAG.

Generating self-repairs incrementally

- Uses same `repair` mechanism as parsing, except trigger is failure to extend the DAG with tree *subsuming* the goal concept.
- Change to goal concept, which is not a subtype of the previous one causes substitution repair.
  - Backtrack along the DAG and try lexical actions again.
- Subtyping of current goal concept after completed utterance will cause abridged repair/extension.

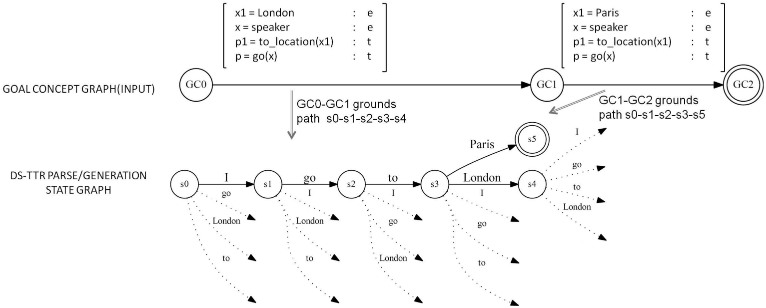
Generating self-repairs incrementally

SYS: "I go to London, uh, Paris"



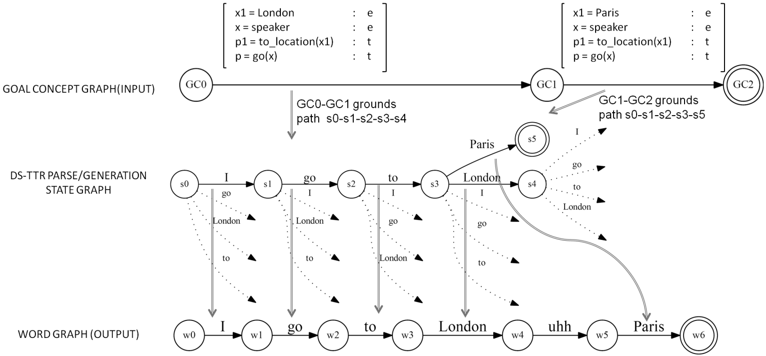
Generating self-repairs incrementally

SYS: "I go to London, uh, Paris"



Generating self-repairs incrementally

SYS: “I go to London, uh, Paris”



All types of self-repair?

- System can deal with direct substitution-type self-repairs and extensions.
- What about:

  "Peter went [swimming **with Susan** + {or rather} surfing], on Tuesday."
- Parsing actions that constructed "with Susan" need to be run again.
- Parsing context DAG to *re-run actions* in the way that DS VP ellipsis works.

Re-running parsing actions

USER: "Peter went swimming with Susan, or rather, surfing..."

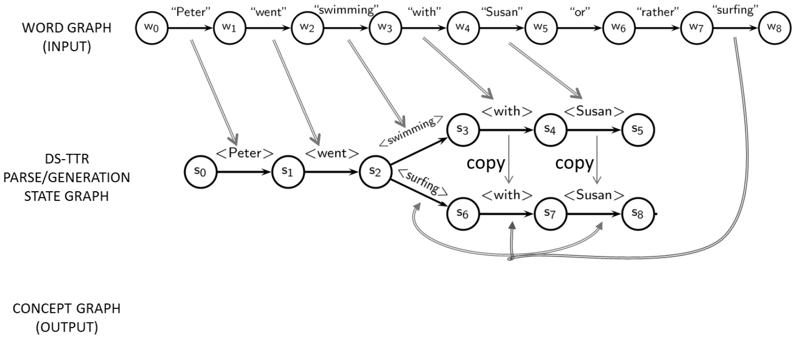
WORD GRAPH (INPUT)

DS-TTR PARSE/GENERATION STATE GRAPH

CONCEPT GRAPH (OUTPUT)

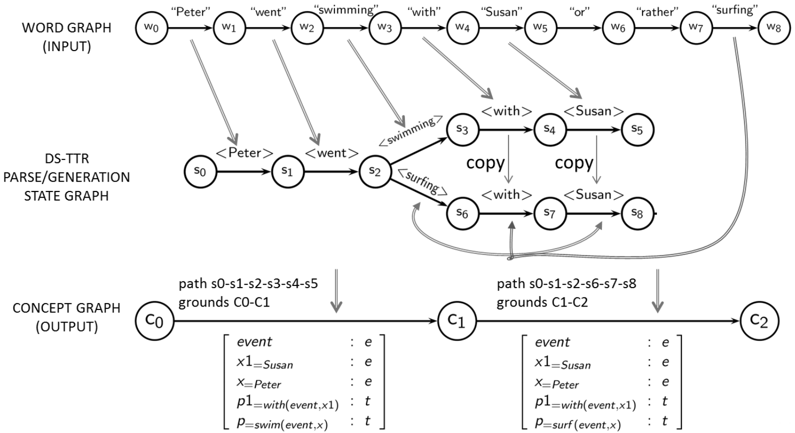
Re-running parsing actions

USER: "Peter went swimming with Susan, or rather, surfing..."



Re-running parsing actions

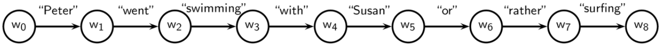
USER: "Peter went swimming with Susan, or rather, surfing..."



Or a TTR solution: Asymmetric merge

USER: "Peter went swimming with Susan, or rather, surfing..."

WORD GRAPH (INPUT)

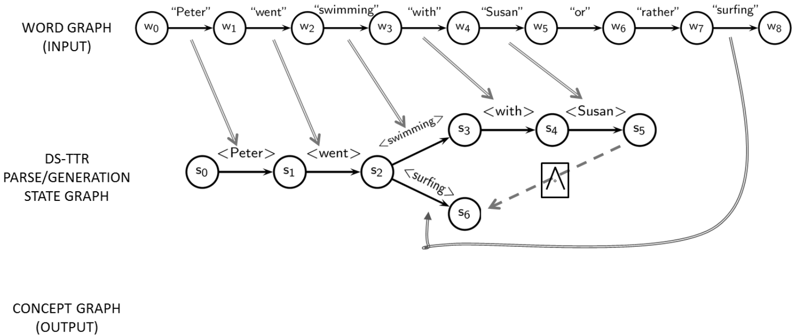


DS-TTR PARSE/GENERATION STATE GRAPH

CONCEPT GRAPH (OUTPUT)

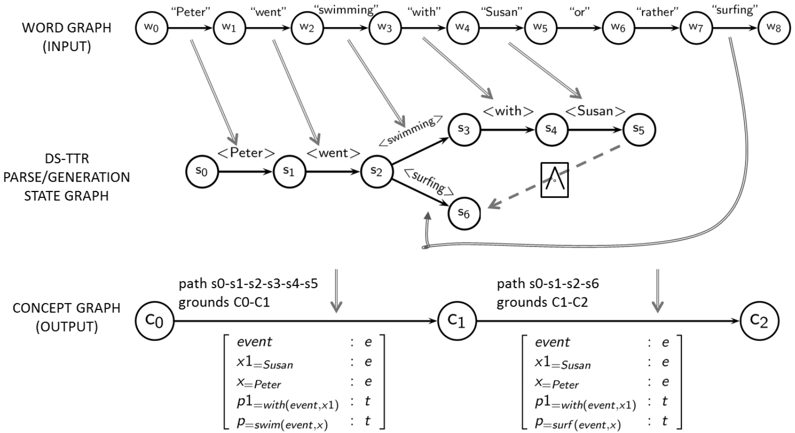
Or a TTR solution: Asymmetric merge

USER: “Peter went swimming with Susan, or rather, surfing...”



Or a TTR solution: Asymmetric merge

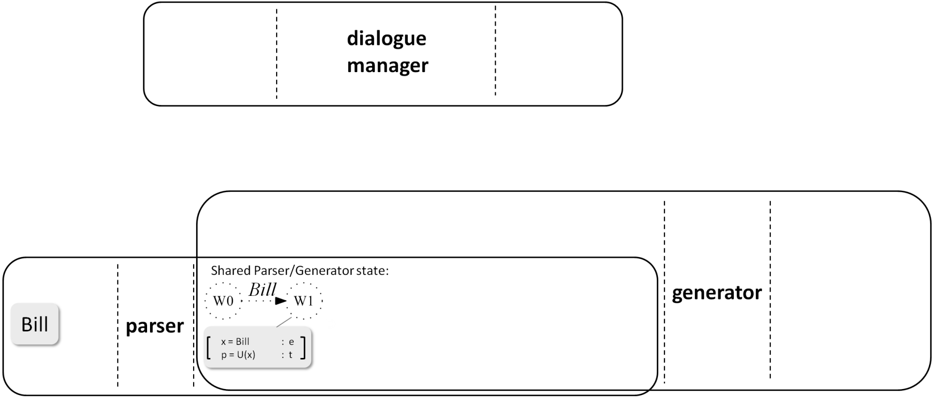
USER: "Peter went swimming with Susan, or rather, surfing..."



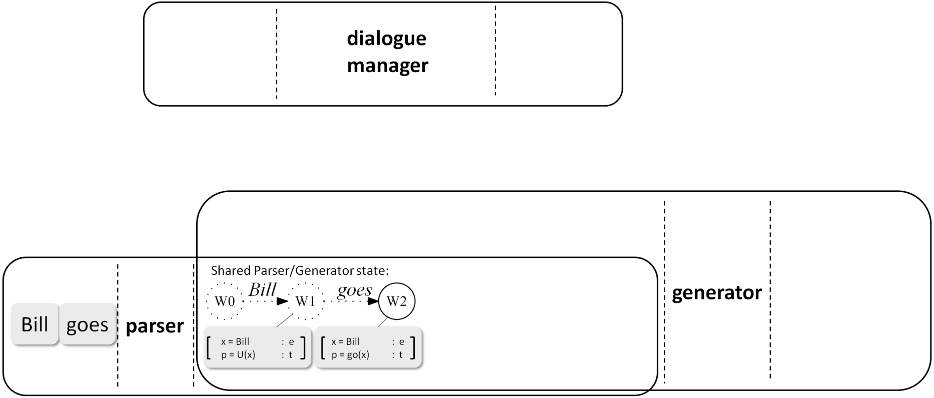
Compound contributions (split utterances) in DyLan

- We can use the common ground/public record [Clark, 1996, Ginzburg, 2012]- parser and generator access the same structure
- Domain concepts in the dialogue manager also record types, provide possible continuations- valid subtypes of the current record type.
- DyLan continually predicts these/restricts the permissible subtypes.

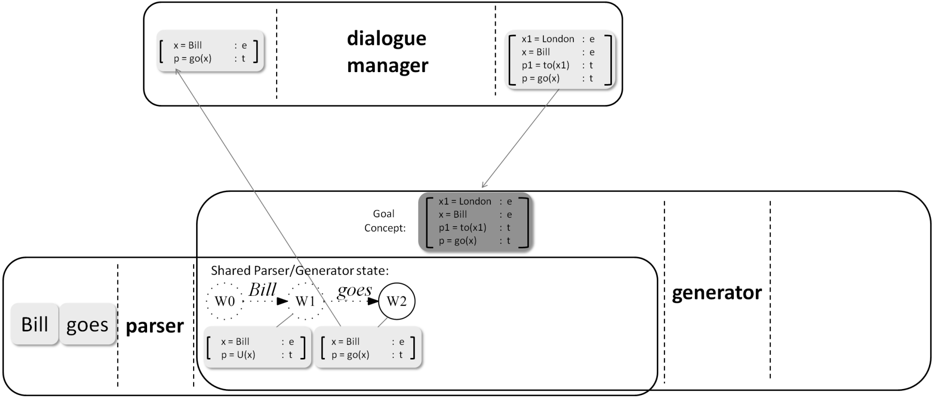
Compound contributions (split utterances) in DyLan



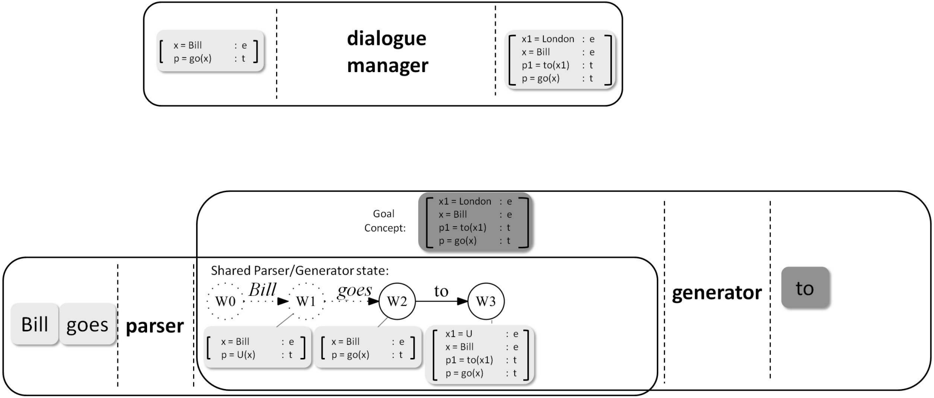
Compound contributions (split utterances) in DyLan



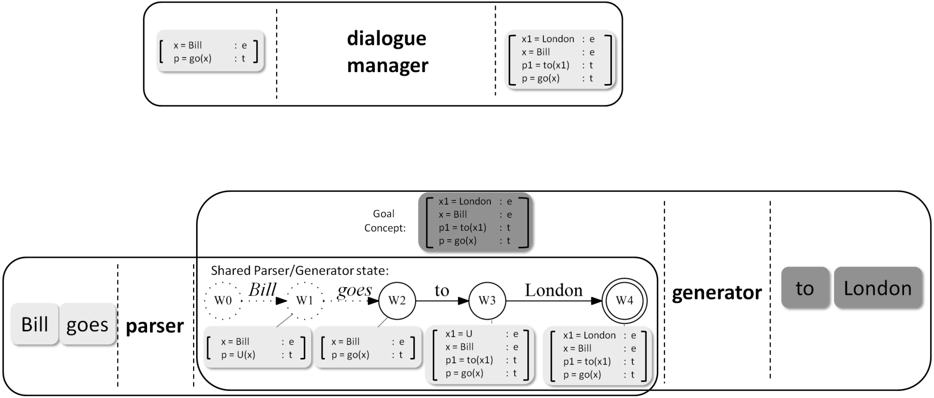
Compound contributions (split utterances) in DyLan



Compound contributions (split utterances) in DyLan



Compound contributions (split utterances) in DyLan



To do

- We can incorporate simple context models, for say indexical resolution, what about integration with more complex dialogue structure (e.g. KOS)?
- Simple subtype selection ok for simple domains...
- In more complex domains how do we constrain the search for goal concepts out of potentially massive space?
- Partial order on record types?
- Probabilities on that order?

Model: Where we're up to

- Incrementally accessible word-by-word semantic representations in TTR record types.
- Incremental *parsing actions* are central to the model and treated as objects.
- Repair now core part of semantic parsing and generation algorithms, recovery as *local* as possible.
- Self-repair can be pragmatically, as well as syntactically constrained.
  - ..and phonologically if we have an incremental ASR.
- Repaired material not discarded and still available in discourse context (level of current *commitment* across graphs.)

Challenges

- Self-repairs: How to best represent relationship between reparandum/repair? Copy-type relations, substitutions etc.
- Compound contributions: How best to organize domain concept record types for fast prediction/selection.
- General challenge: investigate complexity in searching the parsing/generation DAG(s) and the lexicon.

Challenges

- Try different search algorithms, particularly for generation. Heuristics?
- Optimising lexical entry structures for parsing/generation search?
- Evaluation
  - We can talk about complexity order.
  - Scaling up with RISER grammar [Eshghi et al., 2012]
  - Probabilistic TTR [Lappin et al., forthcoming]

Thank you!

Thanks to Matthew Purver, Arash Eshghi, Ruth Kempson, Eleni Gregoromichelaki, Chris Howes, Yo Sato and Robin Cooper among many others.

Questions and suggestions gratefully received...

Brennan, S. and Schober, M. (2001).
How listeners compensate for disfluencies in spontaneous speech* 1.
Journal of Memory and Language, 44(2):274–296.

Buß, O. and Schlangen, D. (2011).
Dium–an incremental dialogue manager that can produce self-corrections.
In *Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 47–54, Los Angeles, California.

Clark, H. H. (1996).
Using Language.
Cambridge University Press.

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

Demberg, V. and Keller, F. (2008).
A psycholinguistically motivated version of tag.
In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms. Tübingen*, pages 25–32.

Gargett, A., Gregoromichelaki, E., Kempson, R., Purver, M., and Sato, Y. (2009).
Grammar resources for modelling dialogue dynamically.
Cognitive Neurodynamics, 3(4):347–363.

Ginzburg, J. (2012).
The Interactive Stance: Meaning for Conversation.
Oxford University Press.

Gregoromichelaki, E., Sato, Y., Kempson, R., Gargett, A., and Howes, C. (2009).
Dialogue modelling and the remit of core grammar.
In *Proceedings of IWCS*.

Guhe, M. (2007).
Incremental Conceptualization for Language Production.
NJ: Lawrence Erlbaum Associates.

Kempen, G. and Hoenkamp, E. (1987).
An incremental procedural grammar for sentence formulation.
Cognitive Science, 11(2):201–258.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001).
Dynamic Syntax: The Flow of Language Understanding.
Blackwell.

Milward, D. (1994).
Dynamic dependency grammar.
Linguistics and Philosophy, 17:561–605.

Milward, D. (1995).
Incremental interpretation of categorial grammar.
In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 119–126. Morgan Kaufmann Publishers Inc.

Neumann, G. (1998).
Interleaving natural language parsing and generation through uniform processing.
Artificial Intelligence, 99:121–163.

Purver, M., Eshghi, A., and Hough, J. (2011).
Incremental semantic construction in a dialogue system.
In Bos, J. and Pulman, S., editors, *Proceedings of the 9th International Conference on Computational Semantics*, pages 365–369, Oxford, UK.

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., and Cann, R. (2010).
Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar.
In Łupkowski, P. and Purver, M., editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, pages 43–50, Poznań. Polish Society for Cognitive Science.

Purver, M. and Kempson, R. (2004).
Incremental context-based generation for dialogue.
In Belz, A., Evans, R., and Piwek, P., editors, *Proceedings of the 3rd International Conference on Natural Language Generation (INLG04)*, number 3123 in Lecture Notes in Artifical Intelligence, pages 151–160, Brockenhurst, UK. Springer.

Sato, Y. (2011).
Local ambiguity, search strategies and parsing in Dynamic Syntax.
In Gregoromichelaki, E., Kempson, R., and Howes, C., editors, *The Dynamics of Lexical Interfaces*, pages 205–233. CSLI.

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.

Shriberg, E. (1994).
Preliminaries to a Theory of Speech Disfluencies.
PhD thesis, University of California, Berkeley.

Shriberg, E. and Stolcke, A. (1998).
How far do speakers back up in repairs? A quantitative model.
In *Proceedings of the International Conference on Spoken Language Processing*, pages 2183–2186.

Skantze, G. and Hjalmarsson, A. (2010).
Towards incremental speech generation in dialogue systems.
In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan. Association for Computational Linguistics.