Optimizing a Discourse Structuring Component for Utterance Generation in Human-Computer Dialogue

Vladimir Popescu, Jean Caelen, Corneliu Burileanu

Constanţa, the 18th of June, 2009

Introduction

- Context: utterance generation in human-computer dialogue systems
- Approaches to utterance generation:
  - rule / grammar-based systems: (i) logic-based; (ii) template-based; (iii) combination
  - corpus-based: (i) purely stochastic; (ii) data-oriented, grammar-based
- Utterance generation components involved in dialogue:
  - surface realization only
  - surface realization + microplanning
  - surface realization + microplanning + discourse structuring
- In focus – a rhetorical structuring component for utterance generation in human-computer dialogue:
  - first-order logic emulation of a fragment of SDRT
  - set of language & task-independent discourse predicates
  - language-independent, domain-specific task ontology
- Roles of the rhetorical structuring component:
  - rhetorical structuring of multi-sentential machine dialogue turns
  - connection of machine dialogue turns to the dialogue history
  - dealing with certain surface phenomena: (i) connectors & other closed-class words (e.g. free choice items); (ii) anaphora & ellipsis, etc.

Baseline Rhetorical Structuring Component

- Usage of SDRT rhetorical relations:
  - Monologic: Alternation, Background, Consequence, Elaboration, Narration, Contrast, Parallel
  - Dialogic: Q-Elab, IQAP, P-Corr, P-Elab, Background$_q$, Elab$_q$, Narration$_q$, QAP, ACK, NEI
- Semantics of rhetorical relations:
  - Expressed in a first-order logic, using the set of **discourse** predicates
  - Preservation of Asher & Lascarides' informal statements
- Examples:
  - Monologic relation: Background(α, β):
    ▹ Semantics: β informs on the *a priori* occurence context of α;
    ▹ Dialogue excerpt (*M* denotes the machine):
    α: *M*: I'm lending you this book, 'X'.
    β: *M*: You have the right to borrow three books at a time.
  - Dialogic relation: P-Corr(α, β) (Plan Correction):
    ▹ Semantics: the emitter of β refutes the plan conveyed by α;
    ▹ Dialogue example (*U* denotes the human user):
    α: *U*: Could I have this book for next week?
    β: *M*: The book is already reserved by another client from the 16th until the 25th.

Baseline Discourse Structure Updating Algorithm

for each utterance α to be added to the dialogue SDRS:

1. read its corresponding logic form $K(\alpha)$, through a query to the *dialogue controller*;
2. for each utterance β already in the dialogue SDRS:
   - **2.1** read its corresponding logic form $K(\beta)$;
   - **2.2** retrieve the set P of potential rhetorical relations (monologic or dialogic);
   - **2.3** for each rhetorical relation ρ in set P:
     - **2.3.1** read the semantics Σ_ρ of rhetorical relation ρ;
     - **2.3.2** compute the truth value γ of the proposition $\Sigma_\rho(K(\alpha), K(\beta))$;
     - **2.3.3** if $\gamma =$ FALSE, then go to step 2.3; else add ρ to the set of rhetorical relations in the SDRS and α to the set of utterances in the SDRS and go to 2.3.

Baseline Algorithm – Complexity Analysis

- Notations:
  - T ::= the time needed to prove a rhetorical relation between two utterances
  - N ::= the number of utterances in the dialogue
  - M ::= the number of speech turns in dialogue
  - R ::= the number of possible rhetorical relations between utterances; = Rd for utterances from different speakers; = Rm for utterances from the same speaker
  - n ::= the index of the current utterance; it varies from 1 to N
  - m ::= the index of the current speech turn; it varies from 1 to M
  - $\upsilon(m)$ ::= the number of utterances in speech turn number m; we have that $\sum_{m=1}^{M} \upsilon(m) = N$ and $\forall m[\upsilon(m) < N - M + 1]$
  - i_m ::= the index of the current utterance in speech turn number m; there are $\upsilon(m)$ values for i_m
  - $\tau_{\text{LAST\_UTT}}$ ::= total running time, for the computation of the rhetorical relations connecting the last utterance to the rhetorical context
- Analysis:

$$\tau_{\text{LAST\_UTT}} = T \times [Rm \times \sum_{i_M=1+\sum_{k=1}^{M-1}\upsilon(k)}^{\sum_{k=1}^{M}\upsilon(k)} (i_M - 1 + \sum_{i\geq 1, M-2i\geq 1} \upsilon(M-2i)) + Rd \times$$

$$\sum_{i_M=1+\sum_{k=1}^{M-1}\upsilon(k)}^{\sum_{k=1}^{M}\upsilon(k)} \sum_{i\geq 0, M-2i\geq 2} \upsilon(M-2i-1)] < (N-M+1) \times T \times Rd \times (3 \cdot N - M - 1) =$$

$$T \times Rd \times (3 \cdot N^2 - 4 \cdot N \times M + M^2 + 2 \cdot N - 1).$$

First Optimization: Usage of Speech Acts

F_U^{FS}:	Where can I find book "X"?	
Possible answers of M		
F_M^S:	It is at the end of this corridor	*QAP*
	Just next to the exit door, to the left	*IQAP*
	The plan of the book shelves is down the entrance hall	*P-Elab*
F_M^{FS}:	Is it for a scientific report you have to write?	$Elab_q$
	Is it a scientific book that you are looking for?	$Background_q$
	This book is to be found in the university library	*P-Corr*
F_M^F:	You go at the end of the corridor	*QAP*
	You go in another building, the "B" building	*P-Corr*
	You go look on the shelf that is just next to the DVD shelves	*IQAP*
F_M^D:	First, you have to validate your card, sir	*P-Elab*
F_M^P:	You can take either the hardcover edition, or the DVD edition	*P-Elab*

Speech Act-based Rhetorical Structuring Algorithm

for each utterance α to be added to the dialogue SDRS:

1. read its corresponding logic form $K(\alpha)$, through a query to the *dialogue controller*;
2. for each utterance β already in the dialogue SDRS:
   - 2.1 read its corresponding logic form $K(\beta)$;
   - 2.2 **read the pair $(\gamma_\alpha, \gamma_\beta)$ of speech acts for this utterance and the utterance at step 1.;**
   - 2.3 **retrieve the set P of rhetorical relations *authorized* by the pair of speech acts read at 2.1;**
   - 2.4 for each rhetorical relation ρ in set P:
     - 2.4.1 read the semantics Σ_ρ of rhetorical relation ρ;
     - 2.4.2 compute the truth value γ of the proposition $\Sigma_\rho(K(\alpha), K(\beta))$;
     - 2.4.3 if γ = FALSE, then go to step 2.4;
       else add ρ to the set of rhetorical relations in the SDRS and α to the set of utterances in the SDRS and go to 2.4.

Speech Act-based Algorithm – Complexity Analysis

- Notations and approximations:
  - $R(n_1, n_2)$ ::= the number of possible rhetorical relations between utterances number n_1 and number n_2; = $Rd(n_1, n_2)$ for utterances from different speakers; = $Rm(n_1, n_2)$ for utterances from the same speaker; on average, $Rd(n_1, n_2) \leq 3$ and $Rm(n_1, n_2) \leq 3$
  - $\tau_{\text{TOT}}^{\text{ACTS}}$ ::= total running time, for the computation of the entire structure, with speech acts;
  - $\tau_{\text{LAST\_UTT}}^{\text{ACTS}}$ ::= total running time, for the computation of the rhetorical relations between the last utterance and the rhetorical context
  - $Rm \mapsto Rm(i_m, i_m - j) : j \in \{1 + \sum_{k=1}^{m-1} \upsilon(k), ..., i_m\} \cup \{2k \text{ such that } i_m - 2k \geq 1\}$
  - $Rd \mapsto Rd(i_m, i_n) : n \neq m \wedge i_m < i_n$
  - $Rm(i_m, i_m - j) \mapsto 3 \wedge Rd(i_m, i_n) \mapsto 3$
- Analysis:
  - $\tau_{\text{TOT}}^{\text{ACTS}} \approx \frac{7}{2} \cdot M \times N^2 \times T \times 3 + M^2 \times N \times (2 - 7 \cdot T \times 3) + M^3 \times (\frac{7}{2} \cdot T \times 3 - 2) + M \times N \times (\frac{7}{2} \cdot T \times 3 - 2) + M^2 \times (4 - \frac{7}{2} \cdot T \times 3) - 2 \cdot M$
  - $\tau_{\text{LAST\_UTT}}^{\text{ACTS}} \approx \frac{1}{6} \cdot \tau_{\text{LAST\_UTT}}$

Second Optimization – Limitation of the Accessible Dialogue History

- Assumptions and notations:
  - speech acts not used $\Rightarrow$ *Rm* candidate rhetorical relations for monologue situations and *Rd* candidates for dialogue
  - Q ::= the maximum number of prior speech turns or utterances to be rhetorically connected to the current one
  - $\tau_{\text{LAST\_TURN}}^{\text{LIMITED\_HISTORY}}$ ::= total running time, for the computation of the rhetorical relations connecting the utterances in the last speech turn to the rhetorical context, with limited dialogue history
- Analysis:

$$\tau_{\text{LAST\_TURN}}^{\text{LIMITED\_HISTORY}} =$$

$$Rm \times T \times \min(\sum_{i_M=1+\sum_{k=1}^{M-1}\upsilon(k)}^{\sum_{k=1}^{M}\upsilon(k)}(i_M - 1 + \sum_{i\geq 1, M-2i\geq 1}\upsilon(M-2i)), \sum_{i_M=1+\sum_{k=1}^{M-1}\upsilon(k)}^{\sum_{k=1}^{M}\upsilon(k)}(i_M -$$

$$1 + \sum_{i\geq 1, M-2i\geq M-Q}\upsilon(M-2i))) + Rd \times T \times \min(\sum_{i_m=1+\sum_{k=1}^{m-1}\upsilon(k)}^{\sum_{k=1}^{m}\upsilon(k)}(\sum_{i\geq 0, m-2i\geq 1}\upsilon(m-2i-$$

$$1)), \sum_{i_m=1+\sum_{k=1}^{m-1}\upsilon(k)}^{\sum_{k=1}^{m}\upsilon(k)}(\sum_{i\geq 0, m-2i\geq M-Q+1}\upsilon(m-2i-1))) \leq Rm \times T \times \sum_{i_M=1+\sum_{k=1}^{M-1}\upsilon(k)}^{\sum_{k=1}^{M}\upsilon(k)}(i_M -$$

$$1 + \sum_{i\geq 1, 2i\leq Q}\upsilon(M-2i)) + Rd \times T \times \sum_{i_m=1+\sum_{k=1}^{m-1}\upsilon(k)}^{\sum_{k=1}^{m}\upsilon(k)}(\sum_{i\geq 0, 2i\leq Q-1}\upsilon(m-2i-1)) <$$

$$T \times Rd \times (2 \cdot \sum_{k=M-Q}^{M-1} + \sum_{k=1}^{M-Q-1} + \frac{\upsilon(M)-1}{2}) \leq T \times Rd \times (-2Q^2 + Q - 1 - 2QM + 2QN + \frac{M+N}{2})$$

Third Optimization – Stack of Computed Discourse Predicates

- Key idea: when a discourse predicate applied to an utterance is evaluated, its truth value is stored $\Rightarrow$ in subsequent uses of the same predicate with the same utterance, the truth value is retrieved (in negligible time), instead of being computed
- Discourse predicates stored in a **stack** $\Leftarrow$ predicates applied to more recent utterances are more likely to be used again $\leftrightarrow$ recency of the available attachment points in discourse structure update

Stack-based Handling of Discourse Predicates

1. each predicate p of a rhetorical relation ρ, applied on the utterance labeled π is computed (e.g. topic(π) , enounce(π)) and used;
2. $p(\pi)$ along with its truth value, denoted by $\theta(p(\pi))$ is stored in the stack, as a pair $p(\pi) \leftrightarrow \theta(p(\pi))$;
3. for a subsequent utterance π', when computing a certain rhetorical relation $\rho(\pi, \pi')$:
   - 3.1 if the predicate $p(\pi)$ belongs to the semantics of ρ, then its truth value is read from the stack and not computed anymore;
   - 3.2 else, the stack is further checked, until either:
     - 3.2.1 a predicate p' belonging to ρ and applied to π' is found: in that case $p'(\pi') \leftrightarrow \theta(p'(\pi'))$ is read from the stack, or
     - 3.2.2 a predicate p' belonging to ρ and applied to π is found: in that case, $p'(\pi) \leftrightarrow \theta(p'(\pi))$ is read from the stack, or
     - 3.2.3 nothing relevant is found in the stack, hence the predicates are computed.

Stack-based Algorithm – Complexity Analysis

- Notations and assumptions:
  - τ ::= the time needed to compute the truth value of a predicate included in the definition of a rhetorical relation; it is assumed constant for all the predicates
  - π_{new} ::= a new utterance, due to be attached to the rhetorical structure
  - Π_{prior} ::= a set of prior utterances, so that $|\Pi_{\text{prior}}| = N$
  - P_{prior} ::= the set of previously used rhetorical relations
  - $\arg(\rho)$ ::= the set of arguments of rhetorical relation ρ
  - $\nu(\rho)$ ::= the number of predicates in the definition of rhetorical relation ρ
  - $\tau^{\text{STACK}}_{\text{LAST\_UTT}}$ ::= the time needed to attach the last utterance to the discourse structure of the dialogue, when the predicate stack is used
  - conjunctions, disjunctions and negations of predicates are assumed to be computed in a negligible time
- Analysis:
  - $\tau_{\text{LAST\_UTT}} = \tau \times \sum_{i=1}^{Rd|Rm} \sum_{\pi \in \Pi_{\text{prior}}} \nu(\rho_i(\pi, \pi_{\text{new}}))$
  - $\tau^{\text{STACK}}_{\text{LAST\_UTT}} = \tau \times \sum_{i=1}^{Rm|Rd} \sum_{\pi \in \Pi_{\text{prior}}} (\nu(\rho_i(\pi, \pi_{\text{new}})) - |\rho_i \cap (\bigcup_{\rho:\rho \in P_{\text{prior}} \wedge \pi \in \arg(\rho)} \rho)|)$
  - $\lim_{|\Pi_{\text{prior}}| \to \infty} (|\rho_i \cap (\bigcup_{\rho:\rho \in P_{\text{prior}} \wedge \pi \in \arg(\rho)} \rho)|) = \nu(\rho_i) \Rightarrow$ for long and *varied* dialogues, the discourse predicate computing time tends to decrease for each rhetorical structure update

A Typical Dialogue

M_1: Welcome to Groplan$^{\pi 11}$. This system allows you to find reference materials in the library of our University$^{\pi 12}$. At any time you can get help by saying general usage mode$^{\pi 13}$. What do you want$^{\pi 14}$?

U_1: Theatre book ancient tragedy humm... ancient Greek theatre$^{\pi 15}$

M_2: I have found more than 13 ancient Greek tragedies$^{\pi 21}$. You can give extra specifications, for instance on the author, or see the solutions or perform a new search$^{\pi 22}$.

U_2: Aristophanes$^{\pi 23}$.

M_3: There is no ancient Greek tragedy written by Aristophanes$^{\pi 31}$. Yet, there are written by Sophocles, Euripides or Aeschylus$^{\pi 32}$. Which proposal are you interested in$^{\pi 33}$?

U_3: A tragedy with parricide I am looking for a tragedy where the son kills his father$^{\pi 34}$. By Aeschylus$^{\pi 35}$.

M_4: There is neither any tragedy with parricide where the son murders his father, written by Aeschylus$^{\pi 41}$. Instead, there are, written by Sophocles or Euripides$^{\pi 42}$. Which proposal are you interested in$^{\pi 43}$?

U_4: I want to read about a son that kills his father$^{\pi 44}$.

M_5: I have 2 tragedies written by Sophocles$^{\pi 51}$. You can see the solutions or perform a new search$^{\pi 52}$.

U_5: See the solutions$^{\pi 53}$.

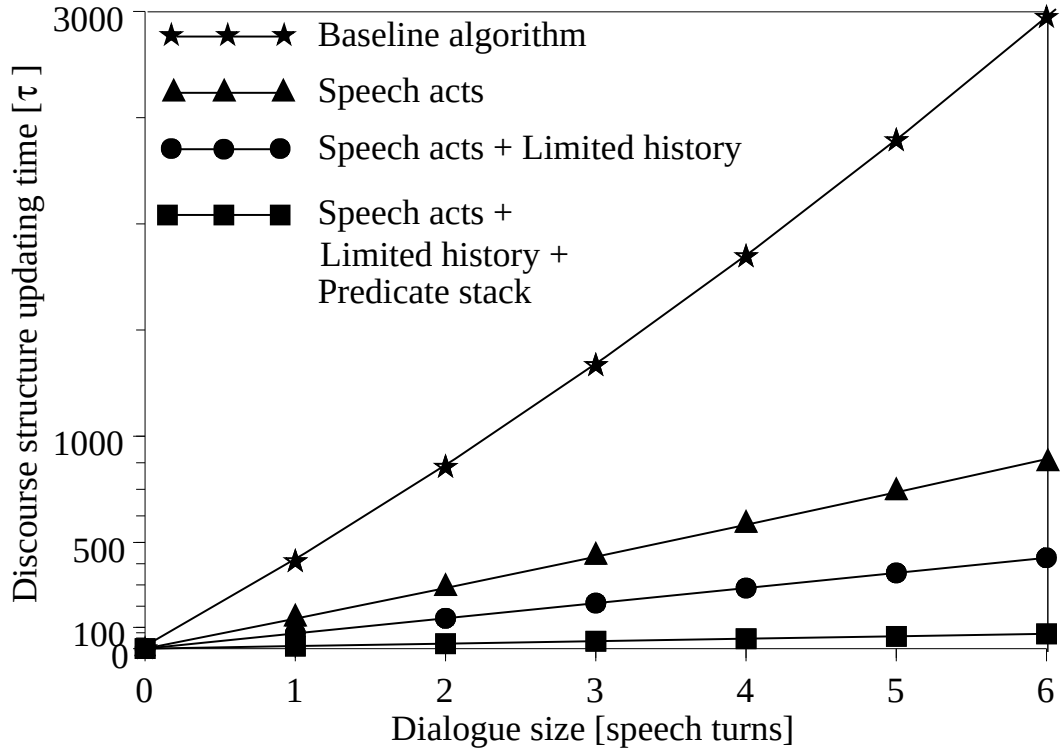
M_6: The first tragedy is “Oedipus the King”$^{\pi 61}$. It is about a prince who murders his father, becomes king and marries his mother, getting to realize and regret it in the end$^{\pi 62}$. Would you like to get more information, the next tragedy or to perform a new search$^{\pi 63}$?

Discourse Structure Updating Times

- SDRT discourse structure:

Backgr(π_{11}, π_{12});
Elab(π_{11}, π_{13});
Conseq($\wedge(\pi_{12}, \pi_{13}), \pi_{14}$);
QAP(π_{14}, π_{15});
P-Elab(π_{15}, π_{21});
P-Elab(π_{15}, π_{22});
Elab(π_{21}, π_{22});
IQAP(π_{22}, π_{23});
P-Corr(π_{23}, π_{31});
Contr(π_{31}, π_{32});
Conseq(π_{32}, π_{33});
IQAP(π_{33}, π_{34});
QAP(π_{33}, Elab(π_{34}, π_{35}));
P-Corr(Elab(π_{34}, π_{35}), π_{41});
Contr(π_{41}, π_{42});
Conseq(π_{42}, π_{43});
IQAP(π_{43}, π_{44});
P-Elab(π_{44}, π_{51});
Conseq(π_{51}, π_{52});
Elab(π_{51}, π_{61});
QAP(π_{52}, π_{53});
ACK(π_{53}, π_{61});
Conseq(Elab(π_{61}, π_{62}), π_{63}).

- Updating Times ($Q \leq 3$):



Conclusions and Prospects

- A series of optimizations brought to a rhetorical structuring component for utterance generation in dialogue systems:
  - baseline algorithm $\leftarrow$ quadratic in the size of the dialogue
  - first optimization $\Leftrightarrow$ speech act usage $\Rightarrow$ constant reduction of the computation time; the same complexity as the baseline
  - second optimization $\Leftrightarrow$ limitation of the dialogue history $\Rightarrow$ complexity reduction: quadratic in the size of the available dialogue history
  - third optimization $\Leftrightarrow$ stack of previously computed discourse predicates $\Rightarrow$ complexity reduction: linear in the size of the dialogue
- The three optimization cascaded $\Rightarrow>$ 100 $\times$ reduction in the discourse updating time with one turn, for a typical 23-utterance dialogue
- Perspective: exhaustive evaluations, either via a Wizard-of-Oz experiment, or via a real dialogue system
- Further prospect: compiling the semantics of the rhetorical relations into abstract attribute grammars (DONE), and embedding them into the updating procedure, for further speedup (TO DO)