Agent utterances in dialogue of cooperative negotiation

No Author Given

No Institute Given

For each utterance chosen by the user, we designed a tree of all the possible responses. These responses take in account the principals of dominance that we aim to model and the current state of the negotiation.

1 State Preference

When the user states a preference that we note StatePreference(C, less,more), where $less, more \in C$. **??** the tree of choices, where each branch depends on the satisfiability of an applicability condition

- **Propose(C,More):** The agent proposes the preferred value. The agent proposes a value only and only if the value is acceptable for the agent. However, the value of acceptability varies depending on the value of dominance.
  – The acceptability of a value is computed with a function that considers the relation of dominance. First, dominant agent is very demanding, while the submissive one is not. Second, the submissive considers the preferences of the user. Indeed, if the user proposes a value that the agent already stated as not preferred, the agent will avoid a conflicting situation and accepts the value. Therefore the function isAcceptable(value, relation) is defined as follow:

1: **function** ISACCEPTABLE($value$, $relation$)
2: **if** ($relation = dominant$) **then**
3: return ($score(value) > bestScoreOfPreference \times 0.7$)
4: **if** ($relation = peer$) **then**
5: return ($score(value) > 0$)
6: **if** ($relation = sub$) **then**
7: return ($score(value) > 0$ or $isInOAS(value)$)

Fig. 1: Function to compute the acceptability of a value of preference

- **StatePreferenc(C,less',more')** : the agent reacts to the stated preference expressed by the user
- **AskPreference(C,less, more)**:

2 AskPreference:

When the user asks the agent about a preference, the agent calls the function reactToUser to compute a preference.

1: **function** REACTTOUSER($less$, $more$)
2: **if** ($more = MostPrefferedValue()$ and $(*, more) \notin OAS$) **then**
3: return $(*, more)$
4: **if** ($more = LeastPrefferedValue()$ and $(more, *) \notin OAS$) **then**
5: return $(more, *)$
6: The conditions $(2, 4)$ are also applied for $less$
7: $(less', more') = computePreference(less, more)$
8: **if** $(less', more') \in OAS$ **then**
9: $(less1, more1) = reactToCriterion(more)$
10: **if** $(less1, more1) = null$ **then**
11: $(less2, more2) = reactToCriterion(less)$
12: **if** $(less2, more2) = null$ **then**
13: return $(less', more')$
14: **else**return $(less2, more2)$
15: return $(less', more')$
16: **function** REACTTOCRITERION($criterion$)

Fig. 2: Function to react to a preference (less,more)

3 Propose(proposal)

the user can either propose a value for a criterion or an option from the set of options of the topic of conversation. Thus, we define a proposal as a tuple $Proposal(Type, Value)$ where $Type$ is either the the topic (for example Restaurants) or a criterion $c \in \mathcal{C}$ and value is:

- an option $O \in\in \mathcal{O}$ if $Type \in Topic$
- a value $v \in D_c$ if $Type \in \mathcal{C}$

- In the case where a user proposes an option that is acceptable for the agent this later calls the task *Booking*, to ask the user to book a table for the restaurant.
- **Accept(Proposal):** the user proposes a value for a criterion that is acceptable for the agent. Thus, the agent expresses an accept.
- **Propose(Proposal'):** The user proposes a proposal which is not acceptable for the agent. This branch is applicable only if the agent is dominant in the relation. A dominant agent is demanding and self centered. Thus, if a user proposes a value which is not acceptable, the agent will counter propose with another value that better suits its preferences.
- **StatePreference(less,more):** The user proposes a proposal which is not acceptable for the agent, and the agent is submissive in its relation with the user. Instead of expressing a reject, the submissive agent will express a statement to express that the proposed value doesn't suit its preferences. In the case of an option, the agent computes the least scored value of this option to explain why he doesn't accept the proposal.
- **Reject(Proposal):** In the same case where the proposed value is not acceptable but the agent is not submissive to express his reject.

4 Accept(proposal)

- **Booking:** When the user accepts an option proposal , the agent closes the negotiation by proposing to book a table for the accepted restaurant (option).
- **Propose(ProposalO):** when a user accepts a criterion proposal, a non-submissive agent continues the negotiation and proposes an option that is defined with the accepted value. However, a submissive agent will only propose an option, if the user accepts a value for each criterion of the topic. The proposed option is defined with all the accepted values.
- after accepting a value for a criterion, the agent opens a negotiation about another criterion of the topic to keep going on the negotiation. Wi distinguish three different responses with respect to the relation of dominance with the user:
  - **StatePreference(less,more)**: A peer agent opens a new subtopic or a negotiation on a criterion by expressing his preferences on this criterion.
  - **AskPreference(less,more)**: A submissive agent considers the preferences of the user and aims to satisfy them. Thus, to open a new negotiation on a criterion, he asks the user about his preferences on this criterion.
  - **Propose(Proposal):** a dominant agent only considers his preferences. Thus, he directs the negotiation to satisfy them, by proposing his most preferred value on the new criterion.

5 Reject(proposal)