Cooperative User Models in Statistical Dialog Simulators

Meritxell González[1,2], **Silvia Quarteroni**[1], **Giuseppe Riccardi**[1], **Sebastian Varges**[1]
[1] DISI - University of Trento, Povo (Trento), Italy
[2] TALP Center - Technical University of Catalonia, Barcelona, Spain
mgonzalez@lsi.upc.edu, name.lastname@disi.unitn.it

Abstract

Statistical user simulation is a promising methodology to train and evaluate the performance of (spoken) dialog systems. We work with a modular architecture for data-driven simulation where the "intentional" component of user simulation includes a User Model representing user-specific features. We train a dialog simulator that combines traits of human behavior such as cooperativeness and context with domain-related aspects via the Expectation-Maximization algorithm. We show that cooperativeness provides a finer representation of the dialog context which directly affects task completion rate.

1 Introduction

Data-driven techniques are a promising approach to the development of robust (spoken) dialog systems, particularly when training statistical dialog managers (Varges et al., 2009). User simulators have been introduced to cope with the scarcity of real user conversations and optimize a number of SDS components (Schatzmann et al., 2006).

In this work, we investigate the combination of aspects of human behavior with contextual aspects of conversation in a joint yet modular data-driven simulation model. For this, we integrate conversational context representation, centered on a Dialog Act and a Concept Model, with a User Model representing persistent individual features. Our aim is to evaluate different simulation regimes against real dialogs to identify any impact of user-specific features on dialog performance.

In this paper, Section 2 presents our simulator architecture and Section 3 focuses on our model of cooperativeness. Our experiments are illustrated in Section 4 and conclusions are summarized in Section 5.

Work partly funded by EU project ADAMACH (022593) and Spanish project OPENMT-2 (TIN2009-14675-C03).

2 Simulator Architecture

Data-driven simulation takes place within the rule-based version of the ADASearch system (Varges et al., 2009), which uses a taxonomy of 16 dialog acts and a dozen concepts to deal with three tasks related to tourism in Trentino (Italy): Lodging Enquiry, Lodging Reservation and Event Enquiry.

Simulation in our framework occurs at the *intention level*, where the simulator and the Dialog Manager (DM) exchange *actions*, i.e. ordered sequences of *dialog acts* and a number of *concept-value* pairs. In other words, we represent the DM action as $a_s = \{da_0, .., da_n\}$, (s is for "System") where da_j is short for a dialog act defined over zero or more concept-value pairs, $da_j(c_0(v_0), .., c_m(v_m))$.

In response to the DM action a_s, the different modules that compose the User Simulator generate an N-best list of simulated actions $A_u(a_s) = \{a_u^0, .., a_u^N\}$. The probability of each possible action being generated after the DM action a_s is estimated based on the conversation context. Such a context is represented by a User Model, a Dialog Act Model, a Concept Model and an Error Model (Quarteroni et al., 2010). The User Model simulates the behavior of an individual user in terms of goals and other behavioral features such as cooperativeness and tendency to hang up. The Dialog Act Model generates a distribution of M actions $A_u = \{a_u^0, .., a_u^M\}$. Then, one action $\hat{a}_u$ is chosen out of A_u. In order to vary the simulation behavior, the choice of the user action $\hat{a}_u$ is a random sampling according to the distribution of probabilities therein; making the simulation more realistic. Finally, the Concept Model generates concept values for $\hat{a}_u$; and the Error Model simulates the noisy ASR-SLU channel by "distorting" $\hat{a}_u$.

These models are derived from the ADASearch dataset, containing 74 spoken human-computer conversations.

2.1 User Model

The User Model represents user-specific features, both transient and persistent. The transient feature we focus on in this work is the user's goal in the dialog (UG), represented as a task name and the list of concepts and values required to fulfill it: an example of UG is {Activity(EventEnquiry), Time_day(2), Time_month(may), Event_type(fair), Location_name(Povo)}.

Persistent features included in our model so far are: patience, silence (no input) and cooperativeness. Patience pat is defined as the tendency to abandon the conversation (hang up event), i.e. $pat = P(HangUp|a_s)$. Similarly, NoInput probability noi is used to account for user behavior in noisy environments: $noi = P(NoInput|a_s)$. Finally, cooperativeness *coop* is a real value representing the ratio of concepts mentioned in a_s that also appear in $\hat{a}_u$ (see Section 3).

2.2 Dialog Act Model

We define three Dialog Act (DA) Models: Obedient (OB), Bigram (BI) and Task-based (TB).

In the Obedient model, total patience and cooperativeness are assumed of the user, who will always respond to each query requiring values for a set of concepts with an answer concerning exactly such concepts. Formally, the model responds to a DM action a_s with a single user action $\hat{a}_u$ obtained by consulting a rule table, having probability 1. In case a request for clarification is issued by the DM, this model returns a clarifying answer. Any offer from the DM to continue the conversation will be either readily met with a new task request or denied at a fixed probability: $A_u(a_s) = \{(\hat{a}_u, 1)\}$.

In the Bigram model, first defined in (Eckert et al., 1997), a transition matrix records the frequencies of transition from DM actions to user actions, including hang up and no input/no match. Given a DM action a_s, the model responds with a list of M user actions and their probabilities estimated according to action distribution in the real data: $A_u(a_s) = \{(a_u^0, P(a_u^0|a_s)), .., (a_u^M, P(a_u^M|a_s))\}$.

The Task-based model, similarly to the "goal" model in (Pietquin, 2004), produces an action distribution containing only the actions observed in the dataset of dialogs in the context of a specific task T_k. The TB model divides the dataset into one partition for each T_k, then creates a task-specific bigram model, by computing $\forall\ k$: $A_u(a_s) = \{(a_u^0, P(a_u^0|a_s, T_k)), .., (a_u^M, P(a_u^M|a_s, T_k))\}$. As the partition of the dataset reduces the number of observations, the TB model includes a mechanism to back off to the simpler bigram and unigram models.

2.3 Concept & Error Model

The Concept Model takes the action $\hat{a}_u$ selected by the DA Model and attaches values and sampled interpretation confidences to its concepts. In this work, we adopt a Concept Model which assigns the corresponding User Goal values for the required concepts, which makes the user simulated responses consistent with the user goal.

The Error Model is responsible of simulating the noisy communication channel between user and system; as we simulate the error at SLU level, errors consist of incorrect concept values. We experiment with a data-driven model where the precision Pr_c obtained by a concept c in the reference dataset is used to estimate the frequency with which an error in the true value $\tilde{v}$ of c will be introduced: $P(c(v)|c(\tilde{v})) = 1 - Pr_c$ (Quarteroni et al., 2010).

3 Modelling Cooperativeness

As in e.g. (Jung et al., 2009), we define cooperativeness at the turn level ($coop_t$) as a function of the number of dialog acts in the DM action a_s sharing concepts with the dialog acts in the user action a_u; at the dialog level, *coop* is the average of turn-level cooperativeness.

We discretize *coop* into a binary variable reflecting high vs low cooperativeness based on whether or not the dialog cooperativeness exceeds the median value of *coop* found in a reference corpus; in our ADASearch dataset, the median value found for *coop* is 0.28; hence, we annotate dialogs as cooperative if they exceed such a threshold, and as uncooperative otherwise. Using a corpus threshold allows domain- and population-driven tuning of cooperativeness rather than a "hard" definition (as in (Jung et al., 2009)).

We then model cooperativeness as two bigram models, reflecting the high vs low value of *coop*. In practice, given a DM action a_s and the *coop* value ($\kappa = high/low$) we obtain a list of user actions and their probabilities: $A_u(a_s, \kappa) = \{(a_u^0, P(a_u^0|a_s, \kappa)), .., (a_u^M, P(a_u^M|a_s, \kappa))\}$.

3.1 Combining cooperativeness and context

At this point, the distribution $A_u(a_s, \kappa)$ is linearly interpolated with the distribution of actions $A_u(a_s, \psi)$ obtained using the DA model ψ (in the Task-based DA model; ψ can have three values, one for each task as explained in Section 2.2):

$$A_u(a_s) = \lambda_\kappa \cdot A_u(a_s, \kappa) + \lambda_\psi \cdot A_u(a_s, \psi),$$

where λ_κ and λ_ψ are the weights of each feature/model and $\lambda_\psi + \lambda_\kappa = 1$.

For each user action a_u^i, λ_κ and λ_ψ are estimated using the Baum-Welch Expectation-Maximization algorithm as proposed by (Jelinek and Mercer, 1980). We use the distributions of actions obtained from our dataset and we align the set of actions of the two models. Since we only have two models, we only need to calculate expectation for one of the distributions:

$$P(\kappa | a_s, a_u^i) = \frac{P(a_u^i | a_s, \kappa)}{P(a_u^i | a_s, \kappa) + P(a_u^i | a_s, \psi)} \; \forall_{i=0}^{M} a_u^i$$

where M is the number of actions. Then, the weights λ_κ and λ_ψ that maximize the data likelihood are calculated as follows:

$$\lambda_\kappa = \frac{\sum_{j=0}^{M} P(\kappa | a_s, a_u^j)}{M}; \lambda_\psi = 1 - \lambda_\kappa.$$

The resulting combined distribution $A_u(a_s)$ is obtained by factoring the probabilities of each action with the weight estimated for the particular distribution:

$$A_u(a_s) = \{(a_u^0, \lambda_\kappa \cdot P(a_u^0 | a_s, \kappa)), .., (a_u^M, \lambda_\kappa \cdot P(a_u^M | a_s, \kappa)),$$

$$(a_u^0, \lambda_\psi \cdot P(a_u^0 | a_s, \psi)), .., (a_u^M, \lambda_\psi \cdot P(a_u^M | a_s, \psi))\}$$

3.2 Effects of cooperativeness

To assess the effect of the cooperativeness feature in the final distribution of actions, we set a 5-fold cross-validation experiment with the ADASearch dataset where we average the λ_κ estimated at each turn of the dialog. We investigated in which context cooperativeness provides more contribution by comparing the λ_κ weights attributed by high vs. low *coop* models to user action distributions in response to Dialog Manager actions.

Figure 1 shows the values achieved by λ_κ for several DM actions for high vs low *coop* regimes. We can see that λ_κ achieves high values in case of uncooperative users in response to DM dialog acts as [ClarificationRequest] and [Info-request]. In contrast, forward-looking actions, such as the ones including [Offer], seem to discard the contribution of the low *coop* model, but to favor the contribution provided by high *coop*.

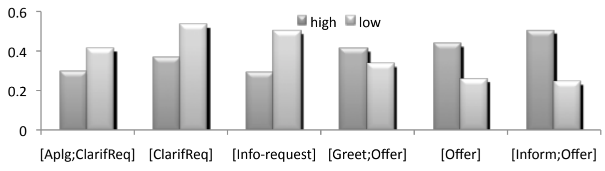

Figure 1: Estimated λ_κ weights in response to selected DM actions in case of high/low *coop*

4 Experiments

We evaluate our simulator models using two methods: first, "offline" statistics are used to assess how realistic the action estimations by DA Models are with respect to a dataset of real conversations (Sec. 4.1); then, "online" statistics (Sec. 4.2) evaluate end-to-end simulator performance in terms of dialog act distributions, error robustness and task duration and completion rates by comparing real dialogs with fresh simulated dialogs using action sampling in the different simulation models.

4.1 "Offline" statistics

In order to compare simulated and real user actions, we evaluate dialog act Precision (P_{DA}) and Recall (R_{DA}) following the methodology in (Schatzmann et al., 2005).

For each DM action a_s the simulator picks a user action $\hat{a}_u$ from $A_u(a_s)$ and we compare it with the real user choice $\tilde{a}_u$. A simulated dialog act is correct when it appears in the real action $\tilde{a}_u$. The measurements were obtained using 5-fold cross-validation on the ADASearch dataset.

Table 1: Dialog Act Precision and Recall

	Simulation (a_u^*)		Most frequent (a_u^*)	
DA Model	P_{DA}	R_{DA}	P_{DA}	R_{DA}
OB	33.8	33.4	33.9	33.5
BI (+*coop*)	35.6 (**35.7**)	35.5 (**35.8**)	49.3 (47.9)	48.8 (47.4)
TB (+*coop*)	38.2 (**39.7**)	38.1 (**39.4**)	51.1 (50.6)	50.6 (50.2)

Table 1 shows P_{DA}/R_{DA} obtained for the OB, BI and TB models alone and with cooperativeness models (+*coop*). First, we see that TB is much better than BI and OB at reproducing real action selection. This is also visible in both P_{DA} and R_{DA} obtained by selecting a_u^*, the most frequent user action from the A_s generated by each model. By definition, a_u^* maximizes the expected P_{DA} and R_{DA}, providing an upper bound for our models; however, to reproduce *any possible* user behavior, we need to sample a_u rather than choosing it by frequency. By now inspecting (+*coop*) values in Table 1, we see that explicit cooperativeness models match real dialogs more closely. It points out that partitioning the reference dataset in high vs low *coop* sets allows better data representation. There is however no improvement in the a_u^* case: we explain this by the fact that by "slicing" the reference dataset, the cooperative model augments data sparsity, affecting robustness.

4.2 "Online" statistics

We now discuss online deployment of our simulation models with different user behaviors and "fresh" user goals and data. To align with the ADASearch dataset, we ran 60 simulated dialogs between the ADASearch DM and each combination of the Task-based and Bigram models and high and low values of *coop*. For each set of simulated dialogs, we measured task duration, defined as the average number of turns needed to complete each task, and task completion rate, defined as: $TCR = \frac{number\ of\ times\ a\ task\ has\ been\ completed}{total\ number\ of\ task\ requests}$.

Table 2 reports such figures in comparison to the ones obtained for real dialogs from the ADASearch dataset. In general, we see that task duration is closer to real dialogs in the Bigram and Task-based models when compared to the Obedient model. Moreover, it can easily be observed in both BI and TB models that under *high-coop* regime (in boldface), the number of turns taken to complete tasks is lower than under *low-coop*. Furthermore, in both TB and BI models, TCR is higher when cooperativeness is higher, indicating that cooperative users make dialogs not only shorter but also more efficient.

Table 2: Task duration and TCR in simulated dialogs with different regimes vs real dialogs.

	Lodging Enquiry		Lodging Reserv		Event Enquiry		All
Model	#turns	TCR	#turns	TCR	#turns	TCR	TCR
OB	9.2±0.0	78.1	9.7±1.4	82.4	8.1±2.9	66.7	76.6
BI+low	15.1±4.1	71.4	14.2±3.9	69.4	9.3±1.8	52.2	66.7
BI+high	**12.1±2.5**	**74.6**	**12.9±3.1**	**82.9**	**7.8±1.8**	**75.0**	**77.4**
TB+low	13.6±4.1	75.8	13.4±3.7	83.3	8.4±3.3	64.7	77.2
TB+high	**11.6±2.8**	**80.0**	**12.6±3.6**	**83.7**	**6.5±1.9**	**57.1**	**78.4**
Real dialogs	11.1±3.0	71.4	12.7±4.7	69.6	9.3±4.0	85.0	73.4

5 Conclusion

In this work, we address data-driven dialog simulation for the training of statistical dialog managers. Our simulator supports a modular combination of user-specific features with different models of dialog act and concept-value estimation, in addition to ASR/SLU error simulation.

We investigate the effect of joining a model of user intentions (Dialog Act Model) with a model of individual user traits (User Model). In particular, we represent the user's cooperativeness as a real-valued feature of the User Model and create two separate simulator behaviors, reproducing high and low cooperativeness. We explore the impact of combining our cooperativeness model with the Dialog Act model in terms of dialog act accuracy and task success.

We find that 1) an explicit modelling of user cooperativeness contributes to an improved accuracy of dialog act estimation when compared to real conversations; 2) simulated dialogs with high cooperativeness result in higher task completion rates than low-cooperativeness dialogs. In future work, we will study yet more fine-grained and realistic User Model features.

References

W. Eckert, E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proc. IEEE ASRU*.

F. Jelinek and R. L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Workshop on Pattern Recognition in Practice*.

S. Jung, C. Lee, K. Kim, and G. G. Lee. 2009. Hybrid approach to user intention modeling for dialog simulation. In *Proc. ACL-IJCNLP*.

O. Pietquin. 2004. *A Framework for Unsupervised Learning of Dialogue Strategies*. Ph.D. thesis, Faculté Polytechnique de Mons, TCTS Lab (Belgique).

S. Quarteroni, M. González, G. Riccardi, and S. Varges. 2010. Combining user intention and error modeling for statistical dialog simulators. In *Proc. INTERSPEECH*.

J. Schatzmann, K. Georgila, and S. Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. SIGDIAL*.

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowl. Eng. Rev.*, 21(2):97–126.

S. Varges, S. Quarteroni, G. Riccardi, A. V. Ivanov, and P. Roberti. 2009. Leveraging POMDPs trained with user simulations and rule-based dialogue management in a spoken dialogue system. In *Proc. SIGDIAL*.