

Hybrid Approach to User Intention Modeling for Dialog Simulation

Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Gary Geunbae Lee

Department of Computer Science and Engineering
Pohang University of Science and Technology(POSTECH)
{hugman, lcj80, getta, gblee}@postech.ac.kr

Abstract

This paper proposes a novel user intention simulation method which is a data-driven approach but able to integrate diverse user discourse knowledge together to simulate various type of users. In Markov logic framework, logistic regression based data-driven user intention modeling is introduced, and human dialog knowledge are designed into two layers such as domain and discourse knowledge, then it is integrated with the data-driven model in generation time. Cooperative, corrective and self-directing discourse knowledge are designed and integrated to mimic such type of users. Experiments were carried out to investigate the patterns of simulated users, and it turned out that our approach was successful to generate user intention patterns which are not only unseen in the training corpus and but also personalized in the designed direction.

1 Introduction

User simulation techniques are widely used for learning optimal dialog strategies in a statistical dialog management framework and for automated evaluation of spoken dialog systems. User simulation can be layered into the user intention level and user surface (utterance) level. This paper proposes a novel intention level user simulation technique.

In recent years, a data-driven user intention modeling is widely used since it is domain- and language independent. However, the problem of data-driven user intention simulation is the limitation of user patterns. Usually, the response patterns from data-driven simulated user tend to be limited to the training data. Therefore, it is not easy to simulate unseen user intention patterns, which is quite important to evaluate or learn optimal dialog policies. Another problem is poor user type controllability in a data-driven method. Sometimes, developers need to switch testers between various type of users such as cooperative, uncooperative or novice user and so on to expose their dialog system to various users.

For this, we introduce a novel data-driven user intention simulation method which is powered by hu-

man dialog knowledge in Markov logic formulation (Richardson and Domingos, 2006) to add diversity and controllability to data-driven intention simulation.

2 Related work

Data-driven intention modeling approach uses statistical methods to generate the user intention given discourse information (history). The advantage of this approach lies in its simplicity and in that it is domain- and language independency. N -gram based approaches (Eckert et al., 1997, Levin et al., 2000) and other approaches (Scheffler and Young, 2001, Pietquin and Dutoit, 2006, Schatzmann et al., 2007) are introduced. There has been some work on combining rules with statistical models especially for system side dialog management (Heeman, 2007, Henderson et al., 2008). However, little prior research has tried to use both knowledge and data-driven methods together in a single framework especially for user intention simulation.

In this research, we introduce a novel data-driven user intention modeling technique which can be diversified or personalized by integrating human discourse knowledge which is represented in first-order logic in a single framework. In the framework, diverse type of user knowledge can be easily designed and selectively integrated into data-driven user intention simulation.

3 Overall architecture

The overall architecture of our user simulator is shown in Fig. 1. The user intention simulator accepts the discourse circumstances with system intention as input and generates the next user intention. The user utterance simulator constructs a corresponding user sentence to express the given user intention. The simulated user sentence is fed to the automatic speech recognition (ASR) channel simulator, which then adds noises to the utterance. The noisy utterance is passed to a dialog system which consists of spoken language understanding (SLU) and dialog management (DM) modules. In this research, the user utterance simulator and ASR channel simulator are developed using the method of (Jung et al., 2009).

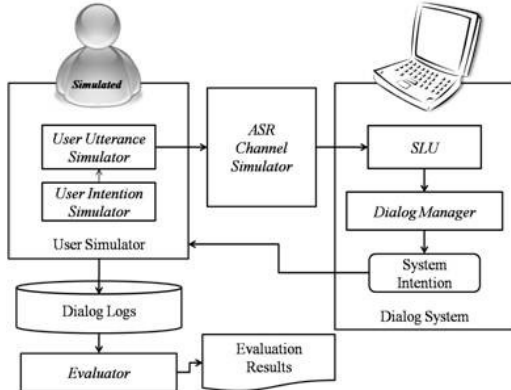


Fig. 1 Overall architecture of dialog simulation

4 Markov logic

Markov logic is a probabilistic extension of finite first-order logic (Richardson and Domingos, 2006). A *Markov Logic Network* (MLN) combines first-order logic and probabilistic graphical models in a single representation.

An MLN can be viewed as a *template* for constructing Markov networks. From the above definition, the probability distribution over possible worlds x specified by the ground Markov network is given by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^F w_i n_i(x)\right)$$

where F is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of F_i in x . As formula weights increase, an MLN increasingly resembles a purely logical KB, becoming equivalent to one in the limit of all infinite weights. General algorithms for inference and learning in Markov logic are discussed in (Richardson and Domingos, 2006).

Since Markov logic is a first-order knowledge base with a weight attached to each formula, it provides a theoretically fine framework integrating a statistically learned model with logically designed and inducted human knowledge. So the framework can be used for building up a hybrid user modeling with the advantages of knowledge-based and data-driven models.

5 User intention modeling in Markov logic

The task of user intention simulation is to generate subsequent user intentions given current discourse circumstances. Therefore, user intention simulation can be formulated in the probabilistic form $P(\text{userIntention} \mid \text{context})$.

In this research, we define the user intention state $\text{userIntention} = [\text{dialog_act}, \text{main_goal}, \text{component_slot}]$, where dialog_act is a domain-independent label of an utterance at the level of illocutionary force (e.g. statement, request, wh_question) and main_goal is the domain-specific user goal of an utterance (e.g. give_something , tell_purpose). Component slots represent domain-specific named-entities in the utterance. For example, in the user intention state for the utterance “*I want to go to city hall*” (Fig. 2), the com-

bination of each slot of semantic frame represents the user intention symbol. In this example, the state symbol is ‘request+search_loc+[loc_name]’. Dialogs on car navigation deal with support for the information and selection of the desired destination.

The first-order language-based predicates which are related with discourse context information and with generating the next user intention are as follows:

- **User intention simulation related predicates**
GenerateUserIntention(context,userIntention)
- **Discourse context related predicates**
hasIntention(context, userIntention)
hasDialogAct(context, dialogAct)
hasMainGoal(context, mainGoal)
hasEntity(context, entity)
isFilledComponent(context,entity)
updatedEntity(context, entity)
hasNumDBResult(context, numDBResult)
hasSystemAct(context, systemAct)
hasSystemActAttr(context, systemActAttr)
isSubTask(context, subTask)

For example, after the following fragment of dialog for the car navigation domain,

```
User(01) : Where are Chinese restaurants?
// dialog_act=wh_question
// main_goal=search_loc
// named_entity[loc_keyword]=Chinese_restaurant
Sys(01) : There are Buchunsung and Idongbanjum in Daeidong.
// system_act=inform
// target_action_attribute=name.address
```

the discourse context which is passed to the user simulator is illustrated in Fig. 3.

Notice that the context information is composed of semantic frame (SF), discourse history (DH) and previous system intention (SI). ‘isFilledComponent’ predicate indicates which component slots are filled during the discourse. ‘updatedEntity’ predicate is true if the corresponding named entity is newly updated. ‘hasSystemAct’ and ‘hasSystemActAttr’ predicate represent previous system intention and mentioned attributes.

raw user utterance	<i>I want to go to city hall.</i>
dialog_act	<i>request</i>
main_goal	<i>search_loc</i>
component.[loc_name]	<i>cityhall</i>

Fig. 2 Semantic frame for user intention simulation on car navigation domain.

SF	hasIntention(“ct_01”, “request+search_loc+loc_name”) hasDialogAct(“ct_01”, “wh_question”) hasMainGoal(“ct_01”, “search_loc”) hasEntity(“ct_01”, “loc_keyword”)
DH	isFilledComponent(“ct_01”, “loc_keyword”) !isFilledComponent(“ct_01”, “loc_address”) !isFilledComponent(“ct_01”, “loc_name”) !isFilledComponent(“ct_01”, “route_type”) updatedEntity(“ct_01”, “loc_keyword”)
SI	hasNumDBResult(“ct_01”, “many”) hasSystemAct(“ct_01”, “inform”) hasSystemActAttr(“ct_01”, “address,name”)

Fig. 3 Example of discourse context in car navigation domain. SF=Semantic Frame, DH=Discourse History, SI=System Intention.

5.1 Data-driven user intention modeling in Markov logic

The formulas are defined between the predicates which are related with discourse context information and corresponding user intention. The formulas for user intention modeling based on logistic regression are as follows:

```

∀ct, pui, ui hasIntention(ct, pui)1
    => GenerateUserIntention(ct, ui)
∀ct, da, ui hasDialogAct(ct, da) => GenerateUserIntention(ct, ui)
∀ct, mg, ui hasMainGoal(ct, mg) => GenerateUserIntention(ct, ui)
∀ct, en, ui hasEntity(ct, en) => GenerateUserIntention(ct, ui)
∀ct, en, ui isFilledComponent(ct, en)
    => GenerateUserIntention(ct, ui)
∀ct, en, ui updatedEntity(ct, en) => GenerateUserIntention(ct, ui)
∀ct, dbr, ui hasNumDBResult(ct, dbr)
    => GenerateUserIntention(ct, ui)
∀ct, sa, ui hasSystemAct(ct, sa) => GenerateUserIntention(ct, ui)
∀ct, attr, ui hasSystemActAttr(ct, attr)
    => GenerateUserIntention(ct, ui)

```

The weights of each formula are estimated from the data which contains the evidence (*context*) and corresponding user intention of next turn (*userIntention*).

5.2 User knowledge

In this research, the user knowledge, which is used for deciding user intention given discourse context, is layered into two levels: *domain knowledge* and *discourse knowledge*. Domain-specific and -dependent knowledge is described in domain knowledge. Discourse knowledge is more general and abstracted knowledge. It uses the domain knowledge as base knowledge. The subtask which is one of domain knowledge are defined as follows

- **Subtask related predicates**
subTaskHasIntention(subTask, userIntention)
moveTo(subtask, subTask)
isCompletedSubTask(context, subTask)
isSubtask(context, subTask)

‘isSubTask’ implies which subtask corresponds to the current context. ‘subTaskHasIntention’ describes which subtask has which user intention. ‘moveTo’ predicate implies the connection from subtask to subtask node.

Cooperative, corrective and self-directing discourse knowledge is represented in Markov logic to mimic following users.

- **Cooperative User:** A user who is cooperative with a system by answering what the system asked.
- **Corrective User:** A user who try to correct the misbehavior of system by jumping to or repeating specific subtask.
- **Self-directing User:** A user who tries to say what he/she want to without considering system’s suggestion.

Examples of discourse knowledge description for three types of user are shown in Fig. 4.

Both the formulas from data-driven model and formulas from discourse knowledge are used for constructing MLN in generation time.

In inference, the discourse context related predicates are given to MLN as true, then probabilities of predicate ‘GenerateUserIntention’ over candidate user intention are calculated. One of example evidence predicates was shown in Fig. 3. All of the predicates of Fig. 3 are given to MLN as true. From the network, the probability of $P(\text{userIntention} \mid \text{context})$ is calculated.

Cooperative Knowledge	
	// If system asks to specify an address explicitly, cooperative users would specify the address by jumping to the address setting subtask.
\forall	ct, st isSubTask(ct, st) ^ hasSystemAct(ct, “specify”) ^ hasSystemActAttr(ct, “address”) => moveTo(st, “AddressSetting”)
Corrective Knowledge	
	// If the current subtask fails, corrective users would repeat current subtask.
\forall	ct, st isSubTask(ct, st) ^ ¬ isCompletedSubTask(ct, st) ^ subTaskHasIntention(st, ui) => GenerateUserIntention(ct, ui)
Self-directing Knowledge	
	// Self-directing users do not make an utterance which is not relevant with the next subtask in their knowledge.
\forall	ct, st isSubTask(ct, st) ^ ¬ moveTo(st, nt) ^ subTaskHasIntention(nt, ui) => ¬ GenerateUserIntention(ct, ui)

Fig. 4 Example of cooperative, corrective and self-directing discourse knowledge.

	A	B	C	D	E	F	G	H
Statistical model (S)	O	O	O	O	O	O	O	O
Cooperative(CPR)		O			O	O		O
Corrective(COR)			O		O		O	O
Self-directing(SFD)				O		O	O	O

Fig. 5 Eight different users (A to H) according to the combination of knowledge.

6 Experiments

137 dialog examples from a real user and a dialog system in the car navigation domain were used to train the data-driven user intention simulator. The SLU and DM are built in the same way of (Jung et al., 2009). After the training, simulations collected 1000 dialog samples at each word error rate (WER) setting (WER=0 to 40%). The simulator model can be varied according to the combination of knowledge. We can generate eight different simulated users from A to H as Fig. 5.

The overall trend of simulated dialogs are examined by defining an *average score* function similar to the reward score commonly used in reinforcement learning-based dialog systems for measuring both a cost and task success. We give 20 points for the successful dialog state and penalize 1 point for each action performed by the user to penalize longer dialogs.

¹ ct: context, ui: user intention, pui: previous user intention, da: dialog act, mg: main goal, en: entity, dbr:DB result, sa: system action, attr: target attribute of system action

Fig. 6 shows that simulated user C which has corrective knowledge with statistical model show significantly different trend over the most of word error rate settings. For the cooperative user (B), the difference is not as large and not statistically significant. It can be analyzed that the cooperative user behaviors are relatively common patterns in human-machine dialog corpus. So, these behaviors can be already learned in statistical model (A).

Using more than two type of knowledge together shows interesting result. Using cooperative knowledge with corrective knowledge together (E) shows much different result than using each knowledge alone (B and C). In the case of using self-directing knowledge with cooperative knowledge (F), the average scores are partially increased against base line scores. However, using corrective knowledge with self-directing knowledge does not show different result. It can be thought that the corrective knowledge and self-directing knowledge are working as contradictory policy in deciding user intention. Three discourse knowledge combined user shows very interesting result. H shows much higher improvement over all simulated users, and the differences are significant results at $p \leq 0.001$.

To verify the proposed user simulation method can simulate the unseen events, the unseen rates of units were calculated. Fig. 7 shows the unseen unit rates of intention sequence. The unseen rate of n-gram varies according to the simulated user. Notice that simulated user C, E and H generates higher unseen n-gram patterns over all word error settings. These users commonly have corrective knowledge, and the patterns seem to not be present in the corpus. But the unseen patterns do not mean poor intention simulation. Higher task completion rate of C, E and H imply that these users actually generate corrective user response to make a successful conversation.

7 Conclusion

This paper presented a novel user intention simulation method which is a data-driven approach but able to integrate diverse user discourse knowledge together to simulate various type of user. A logistic regression model is used for the statistical user intention model in Markov logic. Human dialog knowledge is separated into domain and discourse knowledge, and cooperative, corrective and self-directing discourse knowledge are designed to mimic such type user. The experiment results show that the proposed user intention simulation framework actually generates natural and diverse user intention patterns what the developer intended.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0045).

model \ WER(%)	0	10	20	30	40
A:S (base line)	14.22 (0.00)	9.13 (0.00)	5.55 (0.00)	1.33 (0.00)	-1.16 (0.00)
B:S+CPR	14.39 (0.17)	9.78 (0.65)	5.38 (-0.17)	2.32† (0.99)	-1.00 (0.16)
C:S+COR	14.61† (0.40)	10.91* (1.78)	7.28* (1.74)	2.62‡ (1.30)	-0.81 (0.35)
D:S+SFD	15.70* (1.48)	10.10‡ (0.97)	5.51 (-0.04)	1.89 (0.56)	-0.96* (0.20)
E:S+CPR+COR	14.75‡ (0.53)	10.93* (1.79)	6.88‡ (1.33)	2.94* (1.61)	-1.06† (0.11)
F:S+CPR+SFD	15.75* (1.54)	10.16‡ (1.02)	5.80 (0.26)	1.88 (0.56)	-0.03‡ (1.13)
G:S+COR+SFD	14.39 (0.17)	9.18 (0.05)	5.04 (-0.50)	1.63 (0.31)	-1.52 (-0.36)
H:S+CPR+COR+SFD	15.70* (1.48)	12.19* (3.05)	9.20* (3.65)	5.12* (3.80)	1.32* (2.48)

Fig. 6 Average scores of user intention models over used discourse knowledge. The relative improvements against statistical models are described between parentheses. Bold cells indicate the improvements are higher than 1.0.

† : significantly different from the base line, $p = 0.05$,

‡ : significantly different from the base line, $p = 0.01$,

* : significantly different from the base line, $p \leq 0.001$

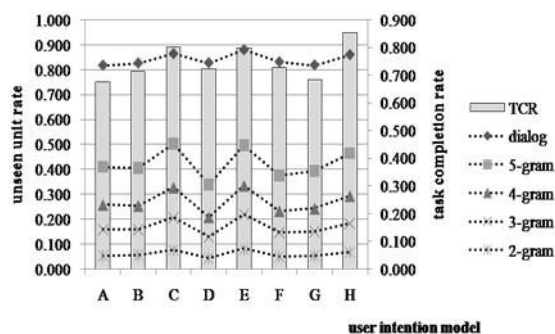


Fig. 7 Unseen user intention sequence rate and task completion rate over simulated users at word error rate of 10.

References

- Eckert, W., Levin, E. and Pieraccini, R. 1997. User modeling for spoken dialogue system evaluation. *Automatic Speech Recognition and Understanding*:80-87.
- Heeman, P. 2007. Combining reinforcement learning with information-state update rules. *NAACL*.
- Henderson, J., Lemon, O. and Georgila, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Comput. Linguist.*, 34(4):487-511.
- Jung, S., Lee, C., Kim, K. and Lee, G.G. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech & Language*.doi:10.1016/j.csl.2009.03.002.
- Levin, E., Pieraccini, R. and Eckert, W. 2000. A stochastic model of human-machine interaction for learning dialog-strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11-23.
- Pietquin, O. and Dutoit, T. 2006. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589-599.
- Richardson, M. and Domingos, P. 2006. Markov logic networks. *Machine Learning*, 62(1):107-136.
- Schatzmann, J., Thomson, B. and Young, S. 2007. Statistical User Simulation with a Hidden Agenda. *SIGDial*.
- Scheffler, K. and Young, S. 2001. Corpus-based dialogue simulation for automatic strategy learning and evaluation. *NAACL Workshop on Adaptation in Dialogue Systems*:64-70.